Integrating Programming Into Mathematics
Math 20

Planning Services

Alberta

Alberta
EDUCATION

Devonian Building, West Tower, 11160 Jasper Avenue, Edmonton, Alberta, Canada T5K 0L2

July 8, 1985

To the Reader:

This study reports the evaluation of a 15 hour Computer Programming in Mathematics (CPM) elective developed and offered in the Math 20 course for two consecutive semesters by five teachers in a large Alberta senior high school.

The educational rationale for the elective is that computer programming in BASIC is in essence a mathematical activity. Once mastered, programming solutions to math concepts in BASIC will result in greater student understanding and achievement in mathematics.

Evaluative results of the initial offerings of the CPM elective seemed to show that it did not meet all of the objectives. However, the elective was viewed as challenging and beneficial by both students and teachers. The findings suggest that the CPM elective is a complex undertaking and one that may not be mastered by a teacher in the first and second offering. Subsequent offerings will likely result in improved student achievement and understanding of fundamental mathematical concepts. Teacher testimony to the worthiness of the elective is also cause for optimism.

Alberta Education is pleased to provide this copy of Integrating Programming into Mathematics 20. It should be of interest to teachers and other educational personnel who are looking for ways to bring computers into the senior high math curriculum.

Sincerely,

H.I. Hastings

H. I. Hastings
Director
Planning Services

INTEGRATING PROGRAMMING INTO MATHEMATICS

MATH 20

Sol E. Sigurdson,
Research Consultant

D. Hunka,
Project Director

A project funded by
Alberta Education
and
Edmonton Public Schools

Acknowledgements

The study owes

-its initiation largely to Tom Kieren, University of Alberta.

-its development to five supportive teachers at Harry Ainlay Composite High School who worked on it for a full school year:

Dan Hunka
Ed Kumish
Mary Parker
Gerry Pruden
Steve Sprysak

- its completion to Somchai Chuchat, graduate student, University of Alberta

The authors are also indebted to the Steering Committee,

Grant Arnold
Art Peddicord
Tom Blowers
Wes Eddy

for its guidance, encouragement and critical comment.

The mathematics departments of

McNally Composite
M. E. LaZerte Composite
Queen Elizabeth Composite
Strathcona Composite

kindly provided control groups for our comparisons.

Finally we are grateful to Planning Services Branch, Alberta Education and Edmonton Public Schools for supporting in many ways this innovative work.

INTEGRATING PROGRAMMING INTO MATHEMATICS

TABLE OF CONTENTS

Introduction.. 1
    The Educational Rationale.................................. 2

Computer Programming in Mathematics........................... 2
    Educational goals of the elective.......................... 3

Curriculum Evaluation - Design................................ 4
    The experimental setting................................... 4
    Goal Evaluation and instruments............................ 5
        Achievement
        Understanding
        Awareness of computers
        Attitude to Computers
        Knowledge of BASIC

Curriculum Evaluation - Analysis and Findings................. 9
    Analysis... 9
    Goal Evaluation - Findings................................. 10
        Achievement
        Understanding
        Awareness of Computers
        Attitude to Computers
        Knowledge of BASIC
    Process Evaluation - Findings.............................. 15

Conclusions and Recommendations............................... 17
    Computer Programming in Mathematics- An Elective...... 17
    Achievement and Understanding.............................. 18
    Computer-related Learnings................................. 19
    Comment.. 19
    Recommendations.. 20

Appendices
    A - Understanding Test..................................21
    B - Computer Awareness..................................24
    C - Nyberg-Clarke Attitude Scales.......................26
    D - Knowledge of BASIC..................................27

INTEGRATING PROGRAMMING INTO MATHEMATICS
MATH 20

Introduction

That computers have a role in mathematics education at any grade level is a given. Wanted are ways that have been worked out for teachers to use them. The elective component of the Math 20 course seemed an ideal way to bring computers into the senior high curriculum. Computers in education is a numbers game - in senior high school, the ratio can be 30 computers to 1400 students. Whatever the ratio, the problem of using computers effectively is a logistical one of assigning students to computers on a one-to-one basis. Apart from this, arises the question of the educational purpose to which computers will be applied. This ranges from problem solving to drill to literacy. Above these considerations is the need for valid instructional programs using computers that the classroom teacher can offer within the context of typical mathematics classrooms.

The current study sought to develop an elective unit in the Math 20 program which would teach students to program, in BASIC, solutions to typical Math 20 exercises. Logistically, the Math 20 class would be scheduled into the computer laboratory for one one-hour period per week. Many alternative approaches to computing in mathematics education are worthwhile. However, with students having computer contacts in other educational settings, programming in a strictly mathematical context seemed the most promising. With this background, the study focussed on three basic questions:

a. Could an instructionally-sound elective unit in Math 20 be developed?

b. In what ways could students benefit from this elective?

c. How did this elective complement the existing Math 20 course?

The Educational Rationale

Why programming when so many alternatives for computer use exist? Programming is an inherently mathematical activity. It deals with variables, functions, and numerical operations. When applied to Math 20 exercises, the activity is especially mathematical. If doing regular exercises in mathematics is practice in using mathematics concepts, then programming solutions to exercises can be thought of as high quality practice. Once the language of BASIC is "mastered," the student's focus is completely on the mathematical concept. Information-processing psychologists would argue that in a highly structured medium such as BASIC, the mathematical concepts can be dealt with more directly, resulting in greater understanding of the mathematics. Clearly this only happens if the student first becomes very familiar with the appropriate level of BASIC language, and secondly if the number of solutions programmed is adequate. Of relevant concern to this study is whether or not the elective can offer a substantial enough programming experience for this benefit to be realized.

Computer Programming in Mathematics(CPM)

The elective unit developed in this study is reported in a separate document- "Computer Programming in Mathematics."(*) The

*______, Math 20, Alberta Education, 1984 development of the elective took place during two semesters in a large senior high school in Alberta which offered Math 20 on a semester timetable. During the first semester, teachers following a course plan taught the elective and met throughout the term to discuss modifications to the original outline. Semester two saw an agreed-upon course being followed by the five teachers. The main guidelines of the elective (presented in detail in the above mentioned document) are:

1. The elective is 15 hours in length.
2. The first five hours are devoted to learning the fundamentals of programming in BASIC.
3. The final ten hours has students programming solutions to typical Math 20 exercises.
4. For one hour per week of the Math 20 course, students are scheduled into the computer lab which contain 30 microcomputers.
5. Teachers offer the elective as an integral part of the Math 20 course with a view to achieving the goals listed below.

The elective as it was offered in the second term was monitored.

Educational goals of the elective

The educational rationale above gives the basic premise under which the elective was offered. Several other goals can be identified. The first four are directly related to the Math 20 course:

1. The student gains a better understanding of mathematical concepts.
2. The use of algorithms and algorithmic thinking are fostered.
3. Fundamental mathematical concepts such as variables, functions, and numerical operations gain new and deeper meaning.

4. Step-by-step, logical, analytical procedures, such as those employed in computer programming, become part of the student's repertoire of mathematical strategies.

The last two goals are more directly related to computers:

5. The student gains an understanding of the computer's role in mathematical activity.
6. The student learns about computers, how they function and, in particular, rudiments of the BASIC language.

Curriculum Evaluation - Design

The focus of the evaluation of any curriculum is on its prescribed goals, determining if students have benefited in predicted ways. While goal evaluation is the primary focus, evaluation of instructional processes is vital to any curriculum evaluation. Typical concerns in instructional process evaluation are the impact of the innovation (the elective) on the regular course (Math 20), teachers' reaction to the required classroom activities, the adequacy of physical facilities, and the appropriateness of instructional materials. The decision to implement any innovation (in this case, the elective) will hinge on the findings of both dimensions of curriculum evaluation.

The experimental setting

The curriculum development and the curriculum evaluation of the CPM elective took place in the first and second term respectively in a large Alberta senior high school. Five Math 20 teachers took part in the project. All five offered the elective in both terms. Evaluation in the second term took the form of a comparison to control groups of Math 20 students in four other senior high schools in the same district. The problem of finding equivalent control groups is enormous since each school has its own standards for allowing students into Math 20. However because the numbers of students involved was large, the control groups were judged to be adequate for purposes of this comparison.

Goal evaluation and instruments

Ideally, goal evaluation attends to all goals of the program. The first four goals, relating directly to Math 20, are accounted for under two evaluation topics: achievement and understanding. The students' achievement and understanding scores should reflect the degree to which the Math 20 goals have been reached. The last two goals relating to computers were dealt with under three areas: computer awareness, attitude to computers, and knowledge of BASIC. Computer awareness looks at computer functioning, capabilities, applications, and societal impact. Attitude simply looks at the student's personal response, while knowledge of BASIC examines student's learning.

Achievement-

Achievement improvement could be affected by the realization of goals 1, 2, and/or 3. But equally important, the 15 hours spend on learning computer programming might have detracted from the usual achievement in Math 20. Both of these aspects of the achievement question can be accounted for by the typical year-end achievement test in Math 20.

Instrument- Math 20 Test

This test covers all aspects of the mathematics curriculum and a range of cognitive levels focussing on the fact, algorithmic, and problem solving levels. It covers all topics in the Math 20 program except the elective and is the usual measure of success in Math 20. The grading of this test was done by the teachers involved using their own scoring schemes for multiple choice and long answer. The total raw score on the test was 103 marks.

Understanding-

This dimension of the evaluation refers especially to goal one but also to goals 2, 3, and/or 4. Understanding in the context of Math 20 was seen as encompassing four aspects:

a. assumptions underlying procedures and formulas

b. analysis of parts of formulas

c. deeper insights into the role of variables, literal coefficients and functions

d. a capacity to deal with questions involving relationships between two or more concepts

Instrument- Understanding

This test(See Appendix A) consisted of 15 questions from topics in the Math 20 course and each relating to one of the above four dimensions of understanding in the following manner:

a. assumptions- 2, 11, 13

b. parts of formulas- 1, 7, 10, 15

c. role of variables, etc.- 5, 6, 8, 12, 14

d. relationships- 3, 4, 9

The grading of the test was based on three marks for each question.

Awareness of computers-

This dimension relates to both goals 4 and 5. The general questions of what computers are, how they are used, and what can we expect of them were asked. Though the CPM elective focussed rather directly on mathematics, application beyond mathematics were discussed formally and informally. The question here is how much benefit accrued in this area. Awareness was taken to encompass six dimensions:

a. the use of languages in communicating to computers
b. the operation of a microcomputer
c. the human element in computer usage
d. the use of computers in mathematical applications
e. the student's concerns about his future with computers
f. societal applications of computers

Instrument- Computer Awareness

The instrument(See Appendix B) consisted of twenty six items with which the student could rate a degree of agreement. The items related to the six dimensions accordingly:

a. language- 1, 12, 18, 20, 26
b. operation- 17, 19, 24, 25
c. human element- 13, 21, 22, 23
d. mathematics- 2, 9, 14, 15, 16
e. student concern- 3, 4, 5
f. society- 6, 7, 8, 10, 11,

Attitude to computers-

The focus was on the students' personal reaction to computers and relates tangentially to goal 5.

Instrument- Nyberg-Clarke Attitude Scale

Although this instrument(See Appendix C) was used to determine the attitude of students to school subjects, it can be applied to attitude toward any subject area, in this instance, computers. The scale is a semantic differential of 24 items, eight of which relate to each dimension of 1) liking the subject, 2) rating its usefulness, and finally 3) difficulty. Each item is given a score of one to five. Every student ends up with three attitude scores, one on each dimension.

Knowledge of BASIC-

In the CPM elective only the rudiments of BASIC were taught. The attempt was made to see if students had learned the language and some elements of simple programs. This relates directly to goal 5.

Instrument- Knowledge of BASIC

The instruments(See Appendix D) consists of 15 multiple choice items centering on the meaning of commands and correct usage of BASIC symbols within a program. Students were given a score of one for each item.

Instructional process evaluation design

Instructional process evaluation examines critically what it means to a teacher to offer this type of program. What thoughts and concerns do teachers have about the CPM program with regard to scheduling, facilities, finances, student motivation, and integration and interaction with the regular program? The researcher visited the five teachers involved in the program several times in their classrooms, interviewed them individually and met with the entire group several times. The instructional process evaluation report based on the data gathered was given to the teachers for their comments. In this way, all the observations have been double checked by the teachers.

Curriculum Evaluation - Analysis and Findings

Analysis

Simple t-tests and chi-square tests were used to compare the project group to the control group. All tests were deemed to be valid when viewed in the light of statistical analysis available for each administration of each instrument. The number of students in each comparison differs and is so stated. The mean for each test or sub-test is given for each group and the 2-tail probability of them being the same. In an exploratory, curriculum-evaluation study of this type, the .1 level of significance, indicated with an asterisk(*), is used. In most of the analyses in this study, we are interested in trends and indications rather than solid proof.

Goal Evaluation- Findings

Achievement

Hypothesis:The mean scores on the Math 20 Test are the same for both the project and control groups.

Analysis: t-test, significance level (.1).

	Project	Control	2-tail Probability
Number of students	133	128	
Total test mean	53	56	.11

Table 1. Math 20 Test - Mean raw score comparison - t-test

Interpretation - Although the difference in achievement is 3 points out of a total of 103, the difference is not significant. The project group did not score higher in achievement. That they did not score significantly lower is also worth noting. In terms of achievement the CPM elective has no affect.

Understanding

Hypothesis: For each of the 15 items on the Understanding Test the mean scores between the project and control groups will be the same.

Analysis: t-test, significance level (.1). Numbers per group as indicated in the Table 2.

Item No.	Project	Control	Significance at .1 level		
*1	1.7	2.5	C		
*2	2.3	1.3	P		
3	.6	.6			
				Project- 41	Control- 43
*4	2.5	1.8	P		
5	2.3	2.2			
6	1.5	1.2			
				Project- 47	Control- 48
7	1.9	1.9			
8	1.6	1.8			
*9	.9	1.4	C		
				Project- 46	Control- 55
*10	1.5	1.9	C		
*11	1.0	1.8	C		
12	1.7	1.6			
				Project- 32	Control- 60
13	1.7	1.9			
*14	.8	1.3	C		
15	.9	1.1			
				Project- 31	Control- 43

Table 2. Understanding Test - Mean score per item comparison - t-test. The numbers for each group are indicated after each three items.

Interpretation - Differences exist in 7 items. Five favour the control while only two favour the project. It is safe to conclude that the project group did not in general score higher on the understanding items. In fact, the opposite is suggested. The items favouring the control group do not fall into any one of the four dimensions of understanding. The seven items seem to follow a pattern related to mathematical topics. For example, the project group did better on both the trigonometry items while the control did better on over half of the quadratic items. This is surprising since many of the computer programming exercises were done on quadratics.

One possible interpretation of these results is that the understanding items, besides measuring understanding, are highly content related. The project group did significantly better in trigonometry and the control in quadratics. But in any case, the results on understanding favour the control group.

Awareness of computers

Hypothesis: 1) For each of the 26 items, the patterns of responses will be the same for each group.

Analysis: 1) A chi-square test for 5 categories was done. The number of students in the project group is 72 and the control, 84. Significance (.1).

Item No.		Difference Indicated
6.	A computer can be programmed to write a play.	More "strongly disagree" from project group.
9.	It is fairly easy to write computer programs to solve math problems.	Fewer "neutral" responses in project group.
10.	Air Canada's computer terminal in Edmonton is is connected to a large central computer somewhere in Canada.	More "strongly agree" responses from project group.
13.	Computer mistakes are usually mistakes made by people.	More "agree" responses from project group.
14.	A computer is useful in solving problems that require creativeness in their solutions	More "strongly disagree" responses from the project group.
15.	A computer is especially useful for tasks that have to be repeated often.	More "strongly agree" responses from the project group.
16.	Computers are good at calculations that require speed, accuracy and repetitiveness.	More "strongly agree" responses from the project group.

Table 3. Computer Awareness - Items showing significant differences - chi square(.1)

Interpretation - For the remaining 19 items in the test no significant differences exist. Four of the 7 items showing differences belong to the "mathematical applications" dimension of the awareness instrument. Interestingly, the project group did not think mathematics programs were easy to write, the main difference was that very few of the project group were undecided. The project group had a greater appreciation of the computer as "processor of numbers" than as a creative problem solver. Two other items indicating a difference belonged the "societal applications" dimensions. Here the project group believed more strongly that a computer cannot write a play and Air Canada has a large central computer. The final item showing a difference is the project group believing that most computer mistakes are people mistakes.

Hypothesis: 2) The mean total awareness score will be the same for each group.

Analysis: 2) After eliminating items 3,4,6,8, and 25; reversing items 5,12,14,17,21, and 23; and assigning a score of 1(strongly disagree) to 5(strongly agree); a mean total awareness score for each group was calculated. A t-test was used to compare the two groups. The numbers of students for the project and control group is 78 and 84.

	Project	Control	2-tail Probability
Number of students	78	84	
Mean total score	85.5	82.7	.03

Table 4. Computer Awareness - Mean total score t-test comparison

Interpretation- Considering all six aspects (dimensions) of computer awareness, the project group is "more aware" of computers than the control. Whether this greater awareness is educationally important is a debatable point. In any case, the control group does appear to know quite a bit about computers but as we have seen from Hypothesis 1) the main difference is not in direction but in the strength of the conviction. In Item 6 for example, both group disagee that a computer can write a play but the project group strongly disagrees.

Attitude to computers

Hypothesis: For each of the three scales of the Nyberg-Clarke Attitude Scales, the mean score is the same both the project and control groups.

Analysis: A t-test was run for each scale separately.
Significance level (.1)
Evaluation- The higher numerical score indicates better liking.
Usefulness- The higher numerical score indicates perceived greater usefulness.
Difficulty- The higher numerical score indicates perceived less difficult.

	Project	Control	2-tail Probability
Number of students	72	84	
Evaluation	27.1	28.7	.12
*Usefulness	32.0	34.8	.00
Difficulty	19.3	20.0	.45

Table 5. Mean score of three scales of the Nyberg-Clarke Attitude Scales - t-test

Interpretation- Interestingly the control is more positive in every scale: like computers more, think they are more useful, and think they are easy. The only statistically significant difference, however, is usefulness. These findings might mean that the project group is less naively optimistic about computers, compared to the general positiveness about them in society. It could also mean that the CPM program has given students a bad feeling about computers.

Knowledge of BASIC

Hypothesis: The mean score on the Knowledge of BASIC Test will be the same for both project and control groups.

Analysis: A t-test was run on the mean total scores. Significance level (.1)

	Project	Control	2-tail Probability
Number of students	73	83	
*Mean Total Score	11.0	9.3	.00

Table 6. Mean total score on Knowledge of Basic Test - t-test.

Interpretation- Students in the project group score about 2 points higher out of a total of 15. The suggests that the average Math 20 class knows a considerable amount of BASIC but that the project group knows more.

Process Evaluation - Findings

The findings of the process evaluation are written into the curriculum and instruction guide of Computer Programming in Mathematics. (*)The reader is referred to this document for an elaboration of the findings. Below is given a short summary:

Elective

1. The CPM elective is a very challenging but rewarding program which most students feel is important.
2. Offering it at some time other than a regular mathematics period, such as noon-hour, has proven problematic.
3. Because of the potential of the computer for many and varied uses in mathematics education, the teacher is encouraged to stick closely to the proposed outline.

*______, Math 20, Alberta Education, 1984

4. Since the benefit of the program is derived once BASIC is learned, the first five lessons must be taught thoroughly.

Students

1. Each Math 20 class contains 4 or 5 students with good computer backgrounds. They did not appear to pose a threat to any of the teachers. Teachers are encouraged to make effective use of them, probably as student helpers both in the classroom and the laboratory.

2. Enrichment activities for advanced students are readily available. Even students with Computing 20 (or similar course) find CPM beneficial.

3. Four or 5 students in any class need special help.

4. Class frustration levels peak during the fourth and fifth lessons when students begin to work seriously at entering and saving programs.

5. Although in general boys are more attracted to computer activity, girls do equally well in CPM.

Programming

1. The first programming assignments should be very explicit.

2. Students need to be encouraged to think programs through before coming to the lab period.

3. Teachers observe that students are consciously and purposefully working with mathematical concepts as they are writing their programs. This, as opposed to the often "rote" approach to the end-of-chapter exercises.

4. The quality of programs range from "bare-bones" to sophisticated. Programs should be written at a level where they can be looked at again after several weeks and easily used. The term "user-friendly" is used in this regard.

5. Students should be encouraged to use their programs to solve problems and to revise previously written programs.

Integration with the regular course

1. Computer programming concepts integrate directly into the regular mathematics lessons. Some examples are
   a. The BASIC command LET is used as a way of assigning variables in problem solutions.
   b. The step-by-step procedures of programming are used in writing up problem solutions.

   c. Literal coefficients which are so prominent in computer programs take on a new status in classroom discussions.

2. Other concepts are indirectly applied
   a. The general concepts of variable and function are reinforced in programming.
   b. The notion of verifying a procedure, important in all mathematics is very explicit in programming.

Facilities

1. Having two or more types of computers in the computer lab adds greatly to the confusion and frustration of lab periods.

2. Teachers should have a computer for use during their preparation time.

Teachers

1. Teachers should know BASIC to the CPM level and be prepared to learn more.

2. Teachers must be completely familiar with the operation of the computer installation in the computer lab.

3. Teachers tended to offer the CPM elective differently, especially in such areas as teaching BASIC, assignment of problems, amount of guidance in programming, grading of programs, expectations on assignments, student-helpers, and carry-over to the regular program.

4. Teachers with a computing background tended to stress the programming component of the CPM elective more.

5. All teachers enjoyed offering the elective and found it stimulated their thinking about mathematics teaching.

Conclusions and Recommendations

Computer Programming in Mathematics- An Elective

The CPM elective as offered was viewed as a challenging program by both teachers and students. Students perceived the elective as an important component of the Math 20 course. Teachers saw the potential for the program to integrate well with the Math 20 course and thought that every student in their classes was benefiting from the experience. Developing the elective in one semester and teaching it again in the second, the teachers in the project are planning to continue improving it and to consolidate it as a well-integrated part of the Math 20 experience.

As the process evaluation findings testify, offering the CPM elective is instructionally a complex undertaking and one that will not be mastered by a teacher in the first or second offering. Factors contributing to this complexity are

1. Teachers gaining some competence in computer programming.
2. The school having appropriate computer facilities which includes scheduling students one-to-one with a computer.
3. Tying the elective to the Math 20 course in a suitable fashion.
4. Accommodating the wide range of student capability in computing.
5. Having students efficiently achieving competence in programming, especially in view of the many other attractions the computer offers.
6. Offering an experience which is so clearly an "elective" to students in the midst of a demanding Math 20 course.

The initial development has been rewarding and indicates a bright future for the CPM elective.

Achievement and Understanding

The evidence here suggests that the CPM program detracted from the regular Math 20 course. If the control groups spent the 15 hours on matters more closely related to the Math 20 program, students would be expected to do better in achievement and understanding. While the differences are not great, we can safely say the gains in these areas do not come automatically with the CPM elective. However, it is expected that an improved offering of the CPM, following this first attempt, will certainly close the gap. Teachers' testimony to the worthwhileness of the elective is also cause for optimism.

Computer-related Learnings

Even here the results do not overwhelmingly favour the elective. Students in regular Math 20 classes know a lot about computers.The CPM students have greater knowledge and awareness of computers but perhaps not to the degree one would expect. Interestingly, the CPM students' attitude to computers is clearly less optimistic (perhaps more realistic) especially in the usefulness dimension. Here, regular Math 20 students perceive computers to be more useful than the CPM students. This might be a consequence of CPM limiting computer experiences to programming. It could, in fact, mean that the typical Math 20 student has an overly optimistic view of the computer's usefulness.

Comment

Although the promise of the CPM elective has not been fulfilled in this particular project, perhaps our sights were initially too high. Should we realistically expect a 15-hour experience to produce measureable effects in level of understanding? Do we know the level of knowledge, awareness or the appropriate attitude for grade eleven students? Probably more important is that teachers and students "sense" that the activity is meaningful and relevant. As one teacher insisted, the CPM should _not_ be an _elective_ it should be part of the course. Math 20 should change to include this experience as an integral part of it. Mathematics courses will change under the impact of computers. The CPM can serve as important groundwork for this exciting prospect.

Recommendations

1. The CPM elective should continue to be offered, improved, and monitored. Particular attention should be paid to
    a. a thorough grounding in rudimentary BASIC
    b. assigning exercises which involve important mathematics concepts from all topics
    c. using their programmed solutions to solve problems or to develop other programs
    d. integrating computer concepts and procedures into the Math 20 curriculum.

2. Teachers should be encouraged to use computers in various ways, outside of the elective, in senior high school mathematics.

3. Studies should be made to find ways of incorporating computers into the mathematics curriculum, especially for students with good computer backgrounds.

4. Studies should be undertaken to investigate classroom activities and testing devices related to the "understanding" dimension of mathematics learning.

5. Consultant and financial resources must be made available before any significant computer usage occurs in classrooms.

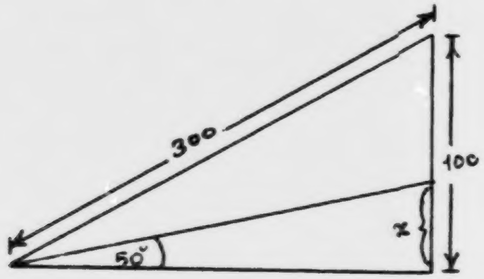
APPENDIX A

Understanding Test

These problems are the problems that were used in the Understanding Test.

1. $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ is the quadratic formula. How does the value of $b^2 - 4ac$ determine what type of roots the quadratic has?

2.



What do you have to assume to be true in order to find the value of x? Why is this assumption necessary? Describe how you would do it.

3. Given 3 points A(3,4) B(5,0) C(1,3), describe how you would find the slope of the line which is the altitude from A.

4. From the top of a cliff which has a height of A metres, a whale is signted by person 2 metres tall at an angle of depression of 20 degrees. How far out to sea is the whale? Do not answer the question.

   Draw a diagram.

   Which trigonometry function would you use to solve the problem?

5. $y = ax^2$

Does the graph of this function open upward or downward? How would you explain this to your classmate who doesn't believe it?

6. For the function $y = mx-b$, if x is increased by 1, how much change do we get in y? Is this an increase or a decrease? Why?

7. Given two equations $y = 2x^2$

$$y = 16x^2$$

Which equation is wider (flatter)? Why does the numerical coefficient make this difference?

8. Given 3 points (a,b) (c,d) (e,f). Describe how you would try to determine if they are all on the same straight line. Do not do the calculation.

9. Given two equations $2x + 3y = -1$

$$2x - 4y = 5$$

You are told to find the equation of a line which passes through the intersection of these two lines and whose slope is 5. Describe how you would do this.

Do not do the algebraic work of actually finding the equation.

10. $f(x) = (x-3)^2 - 5$

Why does $x = 3$ give you the smallest value of $f(x)$?

What value of x gives you the largest value of $f(x)$?

11. The slope of a straight line is constant. What can you say about the slope of a quadratic function?

12. The average height of players in a basketball team is 6 ft. A new player who is 7 ft. tall joins the team. What is the average height of the basketball team now? Describe how you would solve this problem.

13. Given the equation $ax^2 + bx + c = 0$, what are the value of x that make this equation equal to zero?

14. Given the equation $kx^2 - 5 = 0$

How do you find the value of k if you know 2 is a root?

Do not actually calculate the value of k.

15. Given 2 points (2,5) and (10,32) and the segment joining them. How would you find the coordinates of the point 1/3 of the way along the segment? Describe the method. Do not do the actual calculations.

APPENDIX B

COMPUTER AWARENESS

Instruction: There are no right or wrong answers. We want to know what YOU think.

DO NOT PUT YOUR NAME ON THE PAPER.

1. You have to know the language of a computer before you can tell it what to do......................SA A U D SD
2. If you know the rule for solving a mathematics exercise it is easy to do that exercise with a computer..SA A U D SD
3. I can see that I will eventually need to know more about computers.................................SA A U D SD
4. I enjoy playing video games.........................SA A U D SD
5. I am worried about the day when I will have to operate a computer..................................SA A U D SD
6. A computer can be programmed to write a play..........SA A U D SD
7. A computer could be programmed to control the temperature of a school throughout the day............SA A U D SD
8. Most families could use a computer to advantage in their homes......................................SA A U D SD
9. It is fairly easy to write computer programs to solve math problems.................................SA A U D SD
10. Air Canada computer terminal in Edmonton is connected to a large central computer somewhere in Canada...SA A U D SD
11. One computer can do the work of many people...........SA A U D SD

12. Computers can make up their own programs..............SA A U D SD

13. Computer mistakes are usually mistakes made by people...SA A U D SD

14. A computer is useful for solving problems that require creativeness in their solutions..........SA A U D SD

15. A computer is especially useful for tasks that have to be repeated often..........................SA A U D SD

16. Computers are good at calculations that require speed, accuracy and repetitiveness............SA A U D SD

17. The difference between a computer and an electronic typewriter is that a computer has a screen...SA A U D SD

18. The video game "pac-man" is the result of a person writing a program for a computer............SA A U D SD

19. It is possible to program one computer to perform many different tasks..........................SA A U D SD

20. Any computer must be programmed before it can carry out specific tasks or operations............SA A U D SD

21. A computer can determine when someone has put in incorrect information..........................SA A U D SD

22. Information given to a computer must be completely free of errors.............................SA A U D SD

23. A computer quite often makes mistakes................SA A U D SD

24. Special programs can be given to the computer through the disk and the disk drive...................SA A U D SD

25. A computer has a "brain" similar to a person..........SA A U D SD

26. Computers can only do what they are told to do........SA A U D SD

Nyberg-Clarke Attitude Scales

APPENDIX C

Subject Rated

Please place only one mark between each pair of words. Be sure not to leave out any of the pairs.

	very much	a bit	neither	a bit	very much	
nice	===	===	===	===	===	awful
boring	===	===	===	===	===	interesting
unpleasant	===	===	===	===	===	pleasant
dislike	===	===	===	===	===	like
bright	===	===	===	===	===	dull
dead	===	===	===	===	===	alive
lively	===	===	===	===	===	listless (inactive, lazy)
exciting	===	===	===	===	===	tiresome (makes a person feel tired)
useless	===	===	===	===	===	useful
important	===	===	===	===	===	unimportant
impractical	===	===	===	===	===	practical (useful or workable)
worthless	===	===	===	===	===	valuable
helpful	===	===	===	===	===	unhelpful
unnecessary	===	===	===	===	===	necessary
harmful	===	===	===	===	===	advantageous (brings good or gain)
meaningful	===	===	===	===	===	meaningless (doesn't make sense)
hard	===	===	===	===	===	easy
light	===	===	===	===	===	heavy (a lot of work)
clear	===	===	===	===	===	confusing (mixes a person up)
complicated	===	===	===	===	===	simple
elementary	===	===	===	===	===	advanced (beyond the beginning level)
strange	===	===	===	===	===	familiar
understandable	===	===	===	===	===	puzzling (hard to understand)
undemanding	===	===	===	===	===	rigorous (has to be exactly right)

APPENDIX D

Knowledge of BASIC

Instruction: Please circle only one correct answer for each question.

1. What command should you type in order to clear the computer memory?

   a. LIST   b. NEW   c. DEL   d. HOME (or CLR)

2. Why do you have to press the RETURN Key after each statement or command is typed?

   a. To transmit the typed statement or command to the computer memory.

   b. It's the easiest way to have the cursor appear on the left hand side of the screen.

   c. Like a typewriter you can only type one line at a time.

   d. None of the above.

3. What is the different between LIST and RUN?

   a. LIST displays a program, but RUN executes a program.

   b. LIST executes a program, but RUN displays a program.

   c. Both display a program.

   d. Both execute a program.

4. What is the right statement about REM?

   a. REM stands for remark.

   b. A REM statement provides a means of identification for the program.

   c. It does not effect the running of the program.

   d. All of the above.

5. What command should be used to retrieve a program from a disk?

   a. CATALOG b. RECALL

   c. LOAD (or DLOAD) d. SAVE (or DSAVE)

6. What command should be used to show what programs are on a disk?

   a. LIST b. LOAD (or DLOAD)

   c. CATALOG d. SAVE (or DSAVE)

7. Which statement is true about a computer program?

   a. A program is series of instructions.

   b. Each statement needs a line number.

   c. It is executed from the lowest line number.

   d. All of the above.

8. Which operation has the correct notation in BASIC for $\frac{(4\times5)\div2}{10\frac{1}{2}\times8}$

   a. `((4*5)/2)/(10*.5*8)` b. `((4*5)/2)/(10.5*8)`

   c. `(4*5)/(2/10.5)*8` d. `(4*5)/(2/10*1/2*8)`

9. Which sentence is not true for the numeric variable?

   a. The first letter must always be alphabetic.

   b. May be represented by two letters.

   c. May be represented by a letter followed by a single digit.

   d. Must be typed within quotation marks.

10. Let A=2 and B=3. Which pair have the same output in BASIC?

    a. A*B and AB b. A*B and (A)(B)

    c. A*B and (A)*(B) d. AB and (A)(B)

11. Which program contains no error(s)?

a.
```
10 PRINT 6(3+2)
20 END
```

b.
```
10 LET A0 = 6
20 PRINT A0
30 END
```

c.
```
10 LET X = 7
20 LET Y = A*X
30 PRINT Y
40 END
```

d.
```
10 INPUT A
20 LET B = 2+A
30 PRINT B = 2+A
40 END
```

12. Which line of the following program contains error(s)?

```
10 INPUT A
20 LET A+1 = B
30 PRINT B*A
40 END
```

a. line 20   b. line 20 and 30

c. line 30   d. no error line(s)

13. What is the result of running the following program?

```
10 PRINT"A+B = "
20 END
```

a. A+B = A+B   b. A+B = 0

c. A+B = A (or B)   d. A+B =

14. If you run the following program on a computer, what will be the output?

```
10 LET A = 10
20 LET B = 15
30 IF A>B THEN 70
40 LET A = A+10
50 LET B = B+2
60 IF A>=B THEN 80
70 PRINT  A
80 PRINT B
90 END
```

```
a. 17   b. 20   c. 20   d. 10
                   17      15
```

15. What will be the output of the program on the right hand side?

```
10 LET N = 1
20 PRINT N;
30 IF N = 13 THEN 60
40 LET N = N+2
50 GOTO 20
60 END
```

a. 1   b. 1 3 5 7 9 11 13

c. 1 13   d. 1 2 3 4 5 6 7 8 9 10 11 12 13

INTEGRATING PROGRAMMING INTO MATHEMATICS

MATH 20

ALBERTA EDUCATION, 1984

Preface

The document presented here is the result of five senior high school teachers of mathematics working together for a full school year. During this time the elective, Computer Programming in Mathematics, was developed, implemented and monitored. This was a challenging task especially because the development took place within the context of the serious business of teaching grade eleven mathematics. The writer has worked with these teachers, observed their classes, and interviewed them at length. The result is a course outline, a student handbook, and a teachers' guide. The potential for the computer in the teaching of mathematics is great. We hope this document contributes in some way to the realization of this potential. We are pleased to note that everything in these pages has been developed, revised and tested in the classroom under the scrutiny of five demanding teachers. We therefore encourage you to try these materials. It isn't easy but we believe it to be rewarding.

The teacher is referred to the study which reports the extensive monitoring of the elective as it was offered:

> Integrating Programming into Mathematics - Math 20
> Final Report
> Alberta Education, 1984

The teacher is encouraged to monitor his/her own offering of the elective. We have broken new ground here and look forward to you developing and enlarging on our beginnings.

COMPUTER PROGRAMMING IN MATHEMATICS

TABLE OF CONTENTS

STUDENT HANDBOOK COMPUTER FOR PROGRAMMING IN MATHEMATICS 1
- Recommended Outline of Lessons.......................... 2

- Student Manual for BASIC Language........................ 4
  - Lesson 1........................... 5
  - Lesson 2........................... 13
  - Lesson 3........................... 20
  - Lesson 4........................... 25
  - Lesson 5........................... 30
- Answer Key for the Student Manual........................ 36

- Student Exercises for Programming in BASIC - Math 20.... 39

CURRICULUM AND INSTRUCTION GUIDE FOR COMPUTER PROGRAMMING IN MATHEMATICS.............. 48

- The Guide.. 48
- The Goals of the CPM Elective............................ 48
- The CPM Elective... 52
  - Five Lessons in BASIC
  - The first three lessons
  - The fourth lesson
  - The fifth lesson
  - Some variations
- Programming Exercises - Lessons Six to Sixteen.......... 57
  - Three different types of programming tasks
  - Three levels of specificity
  - Assigning exercises
- General Instructional Considerations.................. ... 62
- Enrichment... 65
- CPM as an elective....................................... 66
  - Grading 66
- Relation to the Mathematics Curriculum................... 67
- Student Reaction... 69
- Teacher Preparation...................................... 69
- Teacher Reaction... 70
- The Future of the CPM Elective........................... 71

Appendices... 72
- A- Title Page
- B- Plotting Programs
- C- Typical Grading Schedule
- D- Typical 10 Question Worksheet

SELECTED REFERENCES.. 76

SOLUTION PROGRAMS FOR STUDENT EXERCISES IN PROGRAMMING....... 77

STUDENT HANDBOOK
FOR
COMPUTER PROGRAMMING IN MATHEMATICS

Computer Programming in Mathematics is an elective unit to be offered in the Math 20 course. The 15-hour elective consists of an introduction to programming in BASIC plus instruction in applying this programming knowledge to the solving of mathematics exercises. It does this by limiting the amount of BASIC that the student learns and secondly by limiting the student-written programs to typical end-of-chapter exercises.

This Handbook designed to be given to students consists of:

A. Student Manual for BASIC Language Using the Apple Computer.
This manual is set up in five lessons with practice questions to conclude each lesson.

B. Student Exercises for Programming in BASIC.
These exercises follow the topics of the Alberta Provincial curriculum.

Because there are so many possibilities when computers and mathematics meet and because the Math 20 course only allows 15 hours, the teacher must be careful to adhere to a strict program. For this reason an outline of lessons is given.

Recommended Outline of Lessons

A. Introduction to BASIC

Because the success of this elective depends on students being able to program in BASIC after five one-hour lessons, teachers are encouraged to adhere closely to the suggested outline of lessons.

Lesson One
- Computer elements, function keys
- Deferred and Immediate Mode
- Commands - HOME, NEW, PRINT
- Arithmetic Operations

Lesson Two
- Programming - line numbers
- System commands - LIST, RUN, END, REM
- Variables - A, A2
- Commands - LET, GOTO
- Writing programs - Find the average of three numbers

Lesson Three
- Program analysis of typical programs
- Command - INPUT
- Writing programs using INPUT and GOTO

Lesson Four
- Commands - IF-THEN, FOR-NEXT
- Writing programs using these commands - Print your name 10 times.

Lesson Five
- Disk Commands - SAVE, CATALOG, LOAD, DELETE, RENAME
- Initializing disks
- Writing programs - teacher given programs to be modified
  - Evaluate $2x^2 + 3x + 5$ for $x = 1, 2, 3, \ldots, 10$.
- Saving programs on students' own disks.

Recommended Outline of Lessons (cont'd)

B. Student Exercise for Programming in BASIC

Students should program solutions to two or three exercises in each of the ten periods. To do this, they must have done some planning and thinking about the solution before the computer period begins. Students should save all written programs on their disks. These can be revised and up-dated at a later time.

- Lesson Six and Seven
    - Radicals
    - Polynomials
    - Relations and Functions

- Lesson Eight and Nine
    - Coordinate Geometry

- Lesson Ten
    - Systems of Equations

- Lesson Eleven and Twelve
    - Quadratic Functions

- Lesson Thirteen
    - Variations

- Lesson Fourteen
    - Geometry and Trigonometry

- Lesson Fifteen
    - Statistics

Student Manual for BASIC Language

Using the Apple Computer

page

LESSON 1

1.1 Computer Elements.....................................5
1.2 Function Keys...6
1.3 Immediate and Deferred Modes..........................7
1.4 Utility Commands......................................8
1.5 PRINT Statement.......................................8
1.6 BASIC Operations......................................9
1.7 E Notation...10
Exercise...11

LESSON 2

2.1 Review of Lesson 1...................................13
2.2 Computer Program.....................................13
2.3 System Commands......................................13
2.4 Variables..16
2.5 LET Statement..16
2.6 GOTO Statement.......................................18
Exercise...18

LESSON 3

3.1 Review of Lesson 2...................................20
3.2 INPUT Statement......................................20
3.3 Analyzing Programs...................................21
3.4 Modifying (Re-Writing) Program.......................22
Exercise...23

LESSON 4

4.1 Review of Lesson 3...................................25
4.2 IF-THEN Statement....................................25
4.3 FOR-NEXT Statement...................................26
Exercise...28

LESSON 5

5.1 Review of Lesson 4...................................30
5.2 System Commands to Communicate to the Disk...........30
5.3 Using a Printer......................................33
Exercise...33

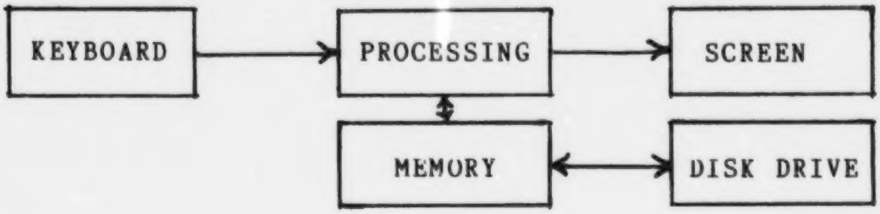
Flow Charting..34

LESSON 1

BASIC, an abbreviation for Beginner's All-purpose Symbolic Instruction Code, is a commonly used computer language. Different computers use different versions of BASIC.

1.1 COMPUTER ELEMENTS

The computer elements are



KEYBOARD provides information to the processing unit.

SCREEN (or printer) displays the processed information or data from the processing unit.

DISK DRIVE (or cassette tape) is used to store information you have in a computer memory onto a disk (or magnetic tape).

PROCESSING occurs in the microprocessing unit (MPU). The MPU obtains a command from Memory and then executes it. Besides adding, subtracting, multiplying, and dividing, the MPU can move information from one spot in the computer to another. In other words, not only can it work with numbers but with letters and symbols as well.

MEMORY is where the MPU finds its instructions and the data it is to work with. There are two main types of memory - Read Only

Memory (ROM) and Random Access Memory (RAM).

ROM: A memory that has a program permanently written into it and cannot be modified by the user. The computer can never write any information into this memory; it is a "read only" memory. ROM is pre-programmed to understand the language (commands) which allow the computer to carry out certain functions. The contents of ROM are not lost when electrical power goes off for any reason.

RAM: A memory system that consists of addresses. These addresses may be "written to or read from" during programming or program execution. Data is placed in specific locations. BASIC statements go into these specific locations when you write a program and are retrieved when you execute the program. Everything in RAM vanishes if electrical power is lost for any reason.

1.2 FUNCTION KEYS

1). CTRL (Control) KEY: TO stop execution of a program hold down the CTRL key and type C. If this doesn't work press CTR key then RESET key.

2). ARROW KEYS:

Right Arrow Key (⟶): The right arrow key moves the cursor one space to the right and rereads any character it has passed over back into computer memory.

Left Arrow Key (⟵): The left arrow key moves the cursor one space to the left and erases each character it has passed over from computer memory, even though the character still appears on the screen.

3). <u>RETURN KEY</u>: The RETURN key must always be pressed after each statement or command is typed. This key transmits the typed statement or command to the computer memory.

1.3 <u>IMMEDIATE AND DEFERRED MODES</u>

Statements may be entered into the microcomputer in either Immediate Mode or Deferred Mode.

1). <u>IMMEDIATE MODE</u>: When a statement is typed into the computer and is executed as soon as the RETURN key is pressed, the computer is said to be in IMMEDIATE MODE. The only way to repeat the statement is to re-type it into the computer. Using the microcomputer in this manner is very similar to using a calculator.

An Example of IMMEDIATE MODE:

```
---------------------------------------------------------
   PRINT 5 + 3 (hit RETURN key)
   8
---------------------------------------------------------
```

2). <u>DEFERRED MODE</u>: Programming occurs in DEFERRED MODE, making it the most frequently used mode. When statements are stored in memory as they are typed in and are executed only when the command RUN is typed, the computer is in DEFERRED MODE. The distinguishing difference between IMMEDIATE and DEFERRED MODES is that statements in DEFERRED MODE are preceded by a line number.

An Example of DEFERRED MODE:

```
---------------------------------------------------------
   10 PRINT"WHAT IS YOUR NAME?" (hit RETURN key)
   RUN (hit RETURN key)
   WHAT IS YOUR NAME?
---------------------------------------------------------
```

1.4 UTILITY COMMANDS

These are basically Immediate Mode commands that are often useful in the execution of a program.

1). HOME: This command clears the display screen and positions the cursor to the upper left-hand corner of the screen. It does not erase a program from memory. In programming, the HOME command is used to start any new display on the screen.

2). NEW: Because the computer stores programs in its memory, you must specifically instruct it to erase an old program from memory before you type in a new program. Do this by typing in NEW then hit RETURN key. If you forget to type NEW, your new program will be mixed in with the old program still in memory. The command NEW erases the contents of RAM. It does not clear the screen nor does it erase anything that is on your disk.

1.5 PRINT STATEMENT

If you want the computer to display something on the screen during program execution, you must issue a PRINT command as part of the program statement. To define what you want the computer to print, type in the PRINT command and enclose whatever you want displayed in quotations (" "). If you want to print the result of a calculation do not use quotes (see line 10 below). You can abbreviate the PRINT command by typing in a question mark (?).

Examples of PRINT:

```
-------------------------------------------------------
    10 LET A=2
    20 LET B=4
    30 LET C=6
    40 PRINT"PROBLEM 2"
    50 PRINT
    60 PRINT A,B,C
    70 PRINT A/2,B/2,C/2
    80 PRINT
    90 PRINT A;B;C
    100 PRINT A+1;B+1;C+1
    110 END
    RUN
    PROBLEM 2

    2               4               6
    1               2               3

    246
    357
-------------------------------------------------------
```

A note for PRINT statement:

1). Anything placed between quotation marks will be printed exactly as typed.

2). A comma tells the computer to divide the line on the screen so that variables or messages are printed 15 spaces apart.

3). A semicolon tells the computer to print the line on the screen with no space between each variable or message printed.

4). A PRINT statement with no messages will leave one blank line in the output.

1.6 BASIC OPERATIONS

BASIC Operations: In BASIC, numbers are operated on by arithmetic operations. These operations must be in a form that understood by the computer. A list of operations and corresponding notation in BASIC is given below.

Operation	BASIC Notation
Addition	+
Subtraction	-
Multiplication	*
Division	/
Raising to a power	^

In BASIC an operations symbol cannot be omitted. For example, the product of A and B must be written A*B, not AB. The order of operations in BASIC is the same as in Algebra.

The following examples illustrate the correct notation of some expressions and also the value of each expression.

Expression	BASIC Notation	Value of Expression
2(8+2)	2*(8+2)	20
$(3-1.5)^2$	(3-1.5)^2	2.25
$\frac{4x9+4}{15-5^2}$	(4*9+4)/(15-5^2)	-4

1.7 E NOTATION

E notation is the computer equivalent of scientific notation. The output of a program in BASIC generally contains a maximum of nine significant digits. When there would be more than nine significant digits, the computer will round off and shift to E notation. Three examples of E notation are given below.

Number	Scientific Notation	E Notation
580,000,000,000	5.8 x 10^{11}	5.8E + 11
4,280,920,616	4.280920616 x 10^{9}	4.28092E + 09
0.0000264280	2.65280 x 10^{-5}	2.64280E - 05

EXERCISE

1. State the commands required by your computer to:

(a) Clear the computer memory.

(b) Clear the screen.

2. On your computer system, what operations are required to:

(a) Transmit a typed statement or command to the computer memory?

(b) Correct a character in a line before the RETURN key is pressed?

(c) Move cursor left one character?

3. List the five arithmetic operations used in BASIC and their BASIC symbols.

4. Each of the following expressions is incorrectly written in BASIC. Rewrite each expression using the correct BASIC notation.

(a) 5X   (b) 184 ÷ 2   (c) 2(X+Y)

(d) 3½*Y   (e) 2 + X^2 + 3X

5. State whether each of the following statements is in Immediate Mode or Deferred Mode.

(a)
```
10 PRINT "(10+5)/3 = ";(10+5)/3 (hit RETURN key)
RUN (hit RETURN key)
(10+5)/3 = 5
```

(b) PRINT "20-15 = ";20-15 (hit RETURN key)

```
20-15 = 5
```

6. Predict output of the following program.

```
10 PRINT"EXERCISE #1"
20 PRINT 2,4,6
30 PRINT"1";"3";"5"
RUN
```

7. Write each of the following numbers using E notation.

(a) 7,600,000,000   (b) 0.001278123   (c) 0.0000000005

LESSON 2

2.1 REVIEW OF LESSON 1

2.2 COMPUTER PROGRAM

A computer program consists of a series of instructions required to solve a specific problem on a computer. The purpose of a computer program is to put instructions and data into the computer (input), have the computer do the execution, and get the results out of the computer (output).

The standard BASIC language consists of about twenty statement types, such as LET, PRINT, and GOTO. Each statement in a BASIC program has a line number which identifies the line and also specifies the order in which the statements are to be executed by the computer. The program below shows how various statements are used.

```
-------------------------------------------------------------
10 LET X = 5                - Assigns 5 to be value of X.
20 LET Y = 7                - Assigns 7 to be value of Y.
30 PRINT X*Y,X+Y            - Computes & prints the values of
40 END                        X*Y and X+Y.
                            - line 40 tells the computer to
                              terminate the program.
-------------------------------------------------------------
```

2.3 SYSTEM COMMANDS

1). LIST: To display a program currently stored in memory, type LIST, then press RETURN key. There are two types of LIST commands. Notice that if the LIST command does not have a line number, then it will list everything in the memory.

An Example of LIST:

```
LIST
10 PRINT"4 + 5 = ";4+5
20 PRINT"8 - 2 = ";8-2
30 PRINT"3 * 9 = ";3*9
40 PRINT"10 / 5 = ";10/5
50 END
```

Sectional listing is also possible. The LIST command followed by one line number will result in that line being listed. A LIST command with two line numbers separated by a comma will result in listing two lines and all the lines between them. For example,

```
LIST 10,30
10 PRINT"4 + 5 = ";4+5
20 PRINT"8 - 2 = ";8-2
30 PRINT"3 * 9 = ";3*9
```

This last sectional listing may also be written LIST 10-30.

2). RUN: This command instructs the computer to execute your program, starting at the lowest line number and terminating when either an END statement, or the highest line number is reached. Type RUN, press RETURN key. Notice that RUN command does not have a line number.

Examples of RUN:

```
10 PRINT"HELLO"
20 PRINT"HOW ARE YOU?"
30 END
RUN
HELLO
HOW ARE YOU?
```

3). END: This command causes a program to cease execution, and returns control to the user.

4). REM: REM stands for remark. A REM statement is put in to remind the programmer of the type of the program. It does not affect the execution of the program. The general form of the REM statement is as follows:

line number REM comment

An Example of REM:

```
-------------------------------------------------------
    10 REM FIND AVERAGE OF THREE TEST SCORES
    20 PRINT"TEST SCORES ARE 4, 6, 8"
    30 PRINT"AVERAGE = ";(4+6+8)/3
    40 REM CALCULATES AND PRINTS AVERAGE
    50 END
-------------------------------------------------------
```

5). DEL: The DEL command is used to delete a line or a group of sequential numbers from a program. A DEL command with two line numbers separated by a comma will result in deleting the two lines and all the lines between them.

An Example of a group of sequential line DELeting:

```
-------------------------------------------------------
    LIST
    10 REM ** FIND THE SQUARES OF 10 AND 20 **
    20 LET N=10
    30 PRINT N,N^2
    40 END
    DEL 20,30
    LIST
    10 REM ** FIND THE SQUARES OF 10 AND 20 **
    40 END
-------------------------------------------------------
```

An Example of one line DELeting:

```
------------------------------------------------------------
    DEL 10,10
    LIST
    40 END
------------------------------------------------------------
```

Another way of deleting a single line in a program is to simply type the line number of the line you want to delete and press RETURN key.

2.4 VARIABLES

A variable is a symbol that is used in programming to represent a number (or other quantities not discussed here.) It refers to that place in the computer memory where the number is stored. In other words, a variable can be thought of as an address that designates a location in the memory. This location contains a value which may vary as the program is being executed.

In BASIC, a variable may be represented by a single letter such as A, B, C,..., Z, or two letters such as AB, AC, BC,..., or a letter followed by a single digit such as A0, A1, A2,..., B0, B1, B2,..., Z9. The first letter of a variable must always be alphabetic and must be typed without quotation marks (" ").

2.5 LET STATEMENT

Specific values can be assigned to a variable by using a LET statement. Consider each of the following programs.

```
------------------------------------------------------------
    10 LET A = 2
    20 LET AB = 8
    30 LET A1 = 5
    40 ? (A + AB)*A1
    50 END
    RUN
    50
------------------------------------------------------------
```

This next program assigns Y a value in terms of the variables X, A, and C.

```
10 LET A = 2
20 LET C = 5
30 LET X = 10
40 LET Y = X*A + C
50 PRINT "Y = ";Y
60 END
```

The equals sign tells the computer to assign the value of the expression on the right to the variable on the left. The left side of the equation must have only one variable. Variables may appear on the right side of the equation as above. However, each variable should be assigned a specific value prior to the statement in line 40.

Consider the following programs.

```
10 LET X = 4              - X and Y are assigned as follows.
20 LET Y = 2                     X = 4        Y = 2
30 LET A = 5*Y + 20/X     - The value of 5*Y+20/X is then
40 PRINT A                  computed and assigned to A in
50 END                      line 30. This value is then
RUN                         printed in line 40.
15                        - Output
```

An Example of LET (with an increment):

```
10 LET X = 1
20 LET X = X + 2
30 PRINT X
40 END
RUN
3
```

The above program shows that the use of the equal sign in programming differs from its use in algebra. (In algebra, X is not equal to X+2).

2.6 GOTO STATEMENT

GOTO is the simplest branching statement; it UNCONDITIONALLY causes program execution to branch to anywhere in the program specified by the GOTO statement. This may be anywhere above or beyond the current point of program execution. Once the GOTO statement has been executed, program execution continues sequentially from the line number indicated in the GOTO statement.

Consider the following program.

```
--------------------------------------------------------
10 LET X = 1
20 LET X = X + 2
30 PRINT X
40 GOTO 20
50 END
RUN
1
3
5
etc.
--------------------------------------------------------
```

Line 40 introduces a GOTO statement. The computer will return to line 20 each time it comes to line 40. You may RUN the above program on your computer to see more output. The GOTO statement has a general form as follows.

line number GOTO line number

EXERCISE

1. State the system commands required by your computer to:
    - (a) Display a program on the screen.
    - (b) Execute a program.
    - (c) Delete a line from a program.
    - (d) Terminate a program.

2. Each of the following programs contains at least one error Find each error and write a correct program.

```
(a) 20 PRINT 6(5+4)    (b) 10 PRINT 3^2    (c) LET A=6
    15 END                 20 END              LET Y=A*3
                           30 RUN              PRINT Y
                                               END
```

3. Determine whether each of the following is an acceptable BASIC numeric variable. Write yes or no.

```
(a) X          (b) XY          (c) A8

(d) 3X         (e) B74         (f) DIF
```

4. State whether each of the following is an acceptable LET statement. Write yes or no.

```
(a) 20 LET A1=6          (b) 20 LET 8=X

(c) 20 LET X=2*X         (d) 20 LET A+B=Z

(e) 20 LET N=R-1         (f) 20 LET X/2=Z/2
```

5. Show the output of each of the following programs.

```
(a) 10 LET N=3           (b) 10 LET T=4
    20 LET Y=5               20 PRINT T,T^2
    30 LET Y=Y+7             30 LET T=T+3
    40 LET X=Y+N             40 PRINT T,T^2
    50 PRINT X               50 LET T=T+1
    60 END                   60 PRINT 3*T
                             70 END
```

6. Each of the following programs contains at least one error. Find each error and write a correct program.

```
(a) LET X=7              (b) 10 LET X=Y+1
    LET Y=X+6                20 LET Y=2
    PRINT X+Y                30 PRINT XY
    END                      40 END

(c) 10 LET X=7
    20 LET Y=A*X
    30 PRINT Y
    40 END
```

LESSON 3

3.1 REVIEW OF LESSON 2

3.2 INPUT STATEMENT

The INPUT statement is another way by which data may be entered into a computer without having to modify the LET statement after each RUN. The INPUT statement is used to obtain information from the user who inputs information into the computer via the keyboard. INPUT can request values for any combination of variables. Once you have correctly entered the information, the RETURN key must be pressed so that the information may be processed and acted upon by the computer.

An Example of INPUT:

```
---------------------------------------------------------------
10 PRINT"ENTER VALUF FOR A";    -Identifies value to enter.
20 INPUT A                      -Requests value for A.
30 PRINT"ENTER VALUE FOR B";    -Identifies value to enter.
40 INPUT B                      -Requests value for B.
50 LET C = A*B                  -Computes the value of A*B
60 PRINT"C = ";C                 and assigns to C. In line
70 END                           60, the value of C is
RUN                              printed.
ENTER VALUE FOR A? 12
ENTER VALUE FOR B? 5
C = 60
---------------------------------------------------------------
```

From the above program, you may rewrite the program as follows.

```
---------------------------------------------------------------
10 PRINT"ENTER VALUES FOR A,B";-Identifies values to enter
20 INPUT A,B                    -Requests values for A & B.
30 LET C = A*B                  -Computes the value of A*B
40 PRINT"C = ";C                 and assigns to C in line
50 END                           30, the value of C is
RUN                              printed.
ENTER VALUES FOR A,B? 12,5
C = 60
---------------------------------------------------------------
```

3.3 ANALYZING PROGRAMS

Please consider the following programs.

(1)

```
------------------
10 LET X = 5
20 LET Y = 10*X + 7
30 PRINT Y
40 END
------------------
```

From the above program at

line 10, the variable X is assigned to have a value of 5.

line 20, the variable Y is assigned to be equal to 10X+7.

line 30, the value of Y is computed then printed.

line 40, END command causes the program to stop execution.

(2)

```
---------------------------------------
10 REM FIND AREA OF A SQUARE,
20 REM GIVEN LENGTH OF SIDE
30 PRINT"GIVE ME THE LENGTH OF A SQUARE"
40 INPUT S
50 PRINT S,S^2
70 END
---------------------------------------
```

From the above program at

line 10 and 20, the REM statements are used to describe the type of the program.

line 30, identifies value to be entered.

line 40, requests value for S (side).

line 50, prints values of length & area of the square.

line 60, tells the computer to restart the program at line 30.

(If you want to stop running of the above program, press CTRL & C or CTRL & RESET keys simultaneously.)

3.4 MODIFYING (RE-WRITING) A PROGRAM

If you want to run the above program (1) many times you could modify it by inserting INPUT and GOTO commands. The results would be

```
------------------------------------------------------------------------
10 INPUT X                  - Requests value for X
20 LET Y = 10*X + 7         - Computes value of 10X+7 and
30 PRINT Y                    assigns to Y. Y is printed.
40 GOTO 10                  - Tells computer to start at
50 end                        line 10 again.
------------------------------------------------------------------------
```

This now means any number can be put in at the INPUT line, and line 40 tells the computer to restart at line 10. The program, in fact will never reach line 50. To stop this program you must press the CTRL & RESET keys simultaneously.

A better program should have line 5 which alerts the person running the program to enter the value for X.

```
---------------------------
5 PRINT"ENTER VALUE FOR X"
10 INPUT X
20 LET Y = 10*X + 7
30 PRINT Y
40 GOTO 5
50 END
---------------------------
```

Now, line 40 tells the computer to restart the program at the "ENTER VALUE FOR X" line

EXERCISE

1. Show the output of the following program.

```
10 HOME
20 PRINT"ENTER BASE & HEIGHT";
30 INPUT B,H
40 LET A=B*H
50 PRINT"BASE","HEIGHT","AREA"
60 PRINT B,H,A
70 GOTO 20
80 END
```

2. The following program contains at least one error. Find each error and write a correct program.

```
10 PRINT"ENTER VALUES A,B";
20 INPUT A;B
30 PRINT"A*B = ";A*B
40 END
```

3. Modify the following program so that a person running the program will know to enter values for A and B. Show the output of program.

```
10 .......
20 INPUT A
30 .......
40 INPUT B
50 PRINT"A+B = ";A+B
60 END
```

4. Fill in the appropriate INPUT statement in lines 20 and 40 of the following program so that the values of A and B can be keyed into the computer memory during execution of the program.

```
10 PRINT"ENTER VALUE FOR A";
20 ...............
30 PRINT"ENTER VALUE FOR B";
40 ...............
50 PRINT A^2;B^2
60 END
```

5. Describe (analyze) what various programs are doing.

```
1. 10 LET X = 25
   20 PRINT X
   30 END
```

```
2. 10 LET Y = 13
   20 LET C = 50
   30 PRINT
   40 PRINT C,Y,C+Y
   50 END
```

```
3.  5 LET M = 5
    7 LET M = M+2
   10 PRINT M
   15 GOTO 7
   25 END
```

```
4. 10 LET X = 37
   20 LET Z = X+4
   30 PRINT X,Z
   40 END
```

```
5. 10 LET A = 5 + 8 + 3 - 2
   20 PRINT A
   30 END
```

```
6. 10 LET Z = 10
   20 PRINT Z
   30 LET Z = Z + 3
   40 PRINT Z
   55 END
```

7. What happens in the above questions when you add

```
50 GOTO 10
```

8. What happens in question 6 when you add

```
50 GOTO 30
```

```
9. 10 LET C = 1
   20 LET D = 3
   25 LET F = C + D
   30 PRINT C, D, E
   40 LET C = C + 1
   50 LET D = D + C
   60 GOTO 25
   70 END
```

```
10. 10 PRINT "TYPE A NUMBER"
    20 INPUT N
    30 PRINT N, 2*N, N*N
    40 END
```

```
11. 10 PRINT "TYPE FIRST NUMBER"
    15 INPUT M
    20 PRINT "TYPE SECOND NUMBER"
    25 INPUT N
    30 LET S = M + N
    40 PRINT "THEIR SUM IS ";S
    50 PRINT
    60 GOTO 10
    70 END
```

```
12. 10 LET N = 13
    20 PRINT N
    30 LET N = N + 11
    40 GOTO 20
    50 END
```

```
13.  10 REM AVERAGE OF 3 NUMBERS
     20 PRINT"TYPE FIRST NUMBER"
     30 INPUT A
     40 PRINT"TYPE SECOND NUMBER"
     50 INPUT B
     60 PRINT"TYPE THIRD NUMBER"
     70 INPUT C
     80 LET M = (A + B + C)/3
     90 PRINT "AVERAGE IS "; M
    100 GOTO 20
    110 END
```

```
14.  10 REM FIND VALUE FOR Y=5X+7
     20 PRINT "ENTER VALUE FOR X"
     30 INPUT X
     40 LET Y = 5*X + 7
     50 PRINT X;",";Y
     60 GOTO 20
     70 END
```

LESSON 4

4.1 REVIEW OF LESSON 3

4.2 IF-THEN STATEMENT

The IF-THEN statement makes a comparison of two numbers. It tells the computer what to do based on the results of the comparison.

The usual form of the IF-THEN statement is as follows.

IF algebraic sentence THEN line number

If the algebraic sentence is true, then the computer proceeds to the line number following THEN. If the algebraic sentence is false the computer simply goes in normal sequence to the next line.

However, instead of a line number after THEN a command may be written. For example,

```
10 IF X < 0 THEN PRINT "NOTE: X IS NEGATIVE"
20 LET Y = 15/X
```

So if the value for X is less than zero the screen will print the message and proceed to line 20. If X is, say, 5, the program will continue to line 20 (without printing anything) and assign Y the value of $15 \div 5$ which is 3. This program, then, would alert the user to negative values of X.

The algebraic sentence in the IF-THEN statement must use one the following symbols.

BASIC Symbol	Meaning
=	is equal to
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to
<>	is not equal to

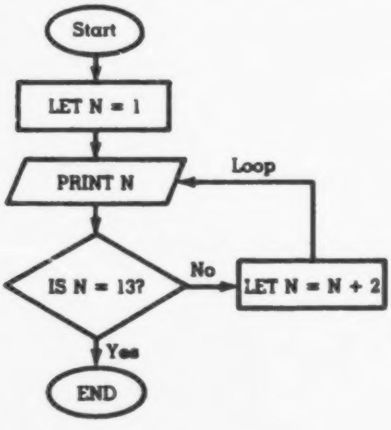
An Example of IF-THEN:

** Write a program which finds the square root of a number. If the number is negative, instruct the computer to print NO REAL SQUARE ROOT. Use INPUT statement.

```
10 PRINT"GIVE ME A NUMBER";
20 INPUT A
30 IF A<0 THEN 60                 -[Test line]
40 PRINT A^.5                     -[Print sq. root of A]
50 GOTO 10                        -[Why is this GOTO
60 PRINT "NO REAL SQUARE ROOT"      statement necessary?]
70 GOTO 10
80 END
```

4.3 FOR-NEXT LOOPS

Loops are often written in a program to direct the computer to repeat a portion of the program a certain number of times. Consider the following flow chart (For a description of the flowchart symbols see Flow Charting, page 34.) and program.



```
10 LET N=1
20 PRINT N;
30 IF N=13 THEN 60
40 LET N=N+2
50 GOTO 20
60 END
```

The same looping procedure can be accomplished by using FOR and NEXT.

```
10 FOR N=1 TO 13 STEP 2 ┐
20 PRINT N;             ├FOR-NEXT loop
30 NEXT N               ┘
40 END
```

A FOR-NEXT loop must begin with a FOR statement and end with a NEXT statement. The variable used in each statement must be the same. Any number of lines may appear between the FOR statement and the NEXT statement.

The general form of the FOR statement is as follows.

<u>FOR</u> Variable = Number <u>TO</u> Number <u>STEP</u> Number

If the step is not indicated, the computer will automatically use a step of one. The second number must be greater than the first number unless the step number is negative. The three numbers may also be variables that have had values assigned to them previously.

When the computer executes a FOR-NEXT loop, the computer automatically tests the value of the variable each time it is incremented. If it is less than or equal to the second number listed in the FOR statement, the computer increases the value by the indicated step and continues in the loop. Otherwise the computer goes to the first statement following the NEXT statement that can be executed.

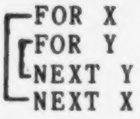
An example of FOR-NEXT

** Write a program that computes Y values for the equation $Y = 8x^2 - 30x + 25$ when x = 1, 1.25, 1.50, 1.75,..., 3.

```
10 PRINT"X","Y"
20 FOR X=1 TO 3 STEP .25
30 LET Y=8*X^2-30*X+25
40 PRINT X,Y
50 NEXT X
60 END
```

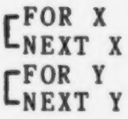
More than one FOR-NEXT loop may occur in a program. There are only two ways that they can appear in a program.

Nested Loops



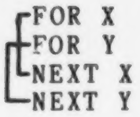
The loops do not cross.

Independent Loops



The loops do not cross. They are not nested.

Not Acceptable



These loops cross.

EXERCISE
----

1. Suppose X=3, Y=10, and Z=15. State what line number the computer will go to after it executes the IF-THEN statement.

```
(a) 30 IF X>2 THEN 50          (b) 40 IF Y<>0 THEN 90
    40 PRINT X                     50 PRINT Y

(c) 30 IF X+Y<Z THEN 70        (d) 50 IF Y^2>Z^2 THEN 70
    40 PRINT X^2+Y^2               60 PRINT Y

(e) 20 IF Y>=12 THEN 40        (f) 20 IF X>Y THEN 50
    30 PRINT X*Y                   30 IF Y<Z THEN 70
                                   40 PRINT X+Z
```

2. Use the partial program at the right to tell whether A, B, or both A and B will be printed. Give the value of any variables printed.

(a) A = 19, B = 5

(b) A = 8, B = 8

(c) A = 13, B = 18

```
10 IF A>B THEN 50
20 LET A=A+10
30 LET B=B+2
40 IF A>=B THEN 60
50 PRINT A
60 PRINT B
70 END
```

3. Write the output of each of the following programs.

```
(a) 10 LET A=1                 (b) 10 LET X=5
    20 PRINT A,A^3                 20 PRINT X
    30 LET A=A+2                   30 LET X=X-2
    40 IF A<7 THEN 20              40 IF X<>-1 THEN 20
    50 END                         50 END
```

(c)
```
10 LET S=0
20 LET N=1
30 LET S=S+N
40 IF N=4 THEN 70
50 LET N=N+1
60 GOTO 30
70 PRINT S
80 END
```

4. Write a FOR statement that assigns the numbers in each of the following lists to the variable X.

(a) 1, 2, 3, 4, 5, 6, 7, 8   (b) 0, 2, 4, 6, 8

(c) 2.25, 2.5, 2.75,...,18   (d) -5, -3.5, -2,..., 9

5. Find the error(s) in each of the following programs and write a correct program. Then show the output of each program.

(a)
```
10 FOR X=1 TO -10
20 NEXT X
30 PRINT X
40 END
```

(b)
```
10 FOR M=3 TO 25,STEP 2
20 PRINT M
30 NEXT M
40 PRINT M^2;M^.5
50 END
```

(c)
```
10 LET X=3
20 FOR C=1 TO 5
30 PRINT C+X
40 NEXT X
50 END
```

(d)
```
10 FOR X=1 TO 6 STEP 3
20 FOR Y=2 TO 4 STEP 2
30 PRINT X*Y
40 NEXT X
50 NEXT Y
60 END
```

6. Please do the following.

(a) Write a program to print your name 10 times.

(b) Write a program to request a name and print that name 10 times.

===

LESSON 5

===

5.1 REVIEW OF LESSON 4

5.2 SYSTEM COMMANDS TO COMMUNICATE TO THE DISK

1). INIT: The INIT command is used to prepare a disk in order that programs can be stored on it. When you purchase a new disk for the purpose of saving programs or data, it is completely blank and therefore will do nothing. Before you can store programs on a disk, it must first be "INITialized". If you want to clear a disk that has programs on it, it must also be "INITialized".

CAUTION! When a disk is INITialized everything stored on it is erased. Make sure you do not INITialize a disk that contains data you want to save.

The INIT command stores on the disk any program that is in memory when you use it. This becomes the "greeting program" which is automatically run every time you put the disk in the disk drive and turn on the power. The "greeting program" can be as simple or complex as you desire. The file name of the greeting program must be specified when you use the INIT command. It is your responsibility to make sure that there is always a program by that name on the disk. If you delete the greeting program, you will see the error message FILE NOT FOUND every time you turn the machine on. The only way to stop that error message is to put a program on that disk with the greeting's file name. That might be difficult to do if you do not know, or cannot remember, what file name was assigned, because there is no way of determining what the name of the greeting program is.

The best solution is prevention. Always specify the same greeting program file name when you initialize your disks. The standard greeting program name is HELLO.

To initialize a new disk first put the System Master Disk in the disk drive then turn the power on, remove it from the drive, and replace it with a new blank disk (or a used disk). Use the NEW command to clear the memory, then type in a greeting program. It is a good idea to test run the greeting program before it is stored on the disk to ensure that it does what you want it to do. Now enter the command: INIT HELLO. Make sure that the drive door is shut and press the RETURN key.

An Example of INIT HELLO:

```
---------------------------------------------------
NEW
10 HOME
20 PRINT"MICHAEL JACKSON...........FEBRUARY 2/84"
30 PRINT"MATH 20-PROGRAMMING EXERCISES"
40 END
---------------------------------------------------
```

2). CATALOG: To see what programs are on the disk, use the CATALOG command. Simply type CATALOG, then press RETURN key and a list of file names will appear on the screen.

3). SAVE: The SAVE command stores programs onto a disk. To store a program you have in computer memory, type in SAVE, a file name followed by a comma and disk drive number then press RETURN key. The file name may consist of up to 30 characters. Every file name must begin with a letter. The disk should whir and clack as the file is being written onto the disk. When SAVE is finished, the BASIC prompt and cursor will reappear on the screen. Type CATALOG to see if your program is listed on the disk directory.

An Example of SAVE:

```
SAVE SQUARE,D1 (hit RETURN key)
```

Since you must type a file name every time you wish to run the program, it is to your advantage to keep it as short as possible. Make sure, however, that the name relates to the program so that at a later date you will have no difficulty remembering its contents.

4). LOAD: The LOAD command loads programs into computer memory. This command does not run the program. To load a program into computer memory, type in LOAD, a file name followed by a comma and disk drive number then press RETURN key.

An Example of LOAD:

```
LOAD SQUARE,D1 (hit RETURN key)
```

5). DELETE: The DELETE command erases files from the disk. Use this command only when you are certain you will no longer require that particular program. To erase a file from your disk, type the DELETE command followed by the file name of the program you wish to erase from your disk.

An Example of DELETE:

```
DELETE SQUARE (hit RETURN key)
```

6). RENAME: The RENAME command changes the name of any file on the disk. To use the RENAME command, type in RENAME, the name of the old file followed by a comma and the new name you wish to give the file. Make sure that the new name you are giving the file has not already been used on that disk.

An Example of RENAME:

```
RENAME SQUARE,DOUBLE (hit RETURN key)
```

This will change the program named SQUARE to DOUBLE.

5.3 USING A PRINTER

To have your program print on the printer.

Load your program into computer memory.

Type PR#1 (hit RETURN key).

Then hold down CTRL key and press I

Type 80N (hit RETURN key)

Ignore the syntax error.

Then type LIST (hit RETURN key).

EXERCISE

1. State the system commands required by your computer to:
   (a) Store a program onto a disk.
   (b) Display a list of file names on the screen.
   (c) Change a program name.
   (d) Remove a program from a disk.

2. If you want to store a program named "PROBLEM #1" on to your disk, what format of command you should type in?

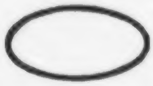
3. To remove a program named "EXERCISE #2" from your disk, what format of command you should type in?

Flow Charting

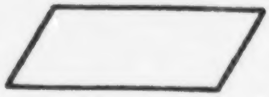
A flow chart is a diagram that shows a step-by-step procedure for solving the problem. Its purpose is to clearly define an overall plan so that you can more readily visualize the logical flow of the program.

The shapes which are used to draw flow charts have special meanings.

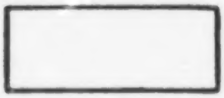
An oval is used to begin or end a program.



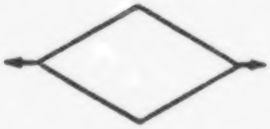
A parallelogram shows input or output. It is used with READ or PRINT statements.



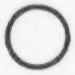
A rectangle shows processing operations. It is used with LET statements.



A diamond shows a decision. Arrows show how the flow continues. It is used with IF-THEN statements.

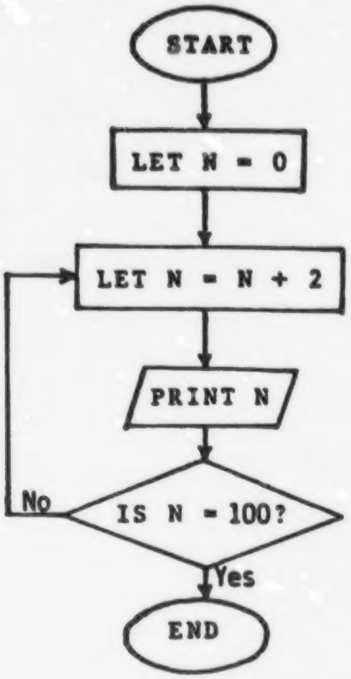


A circle is used to connect shapes when drawing an arrow is inconvenient. With an appropriate letter or number inside, one circle denotes an exit from one part of the flow chart, while a second circle indicates the entry to



another part of the flow chart, thus maintaining a continuous flow of logic.

Consider the following flow chart of the program on the right which prints the even numbers from 2 to 100 inclusive and stops at number 100.


START
LET N = 0
LET N = N + 2
PRINT N
No
IS N = 100?
Yes
END

```
10 LET N=0
20 LET N=N+2
30 PRINT N
40 IF N=100 THEN 60
50 GOTO 20
60 END
RUN
2
4
6
.
.
.
100
```

Answer Key for the Student Manual

Exercise 1

1. a) NEW   b) HOME

2. a) Press RETURN key

   b) Move cursor to that position and correct the character

   c) Press the Left Arrow key

3.

Addition	+
Subtraction	-
Multiplication	*
Division	/
Raising to a power	^

4. a) 5*X   b) 184/2   c) 2*(X+Y)   d) 3.5*Y   e) 2+X^2+3*X

5. a) Deferred Mode   b) Immediate Mode

6.
```
EXERCISE #1
2      4      6
1 3 5
```

7. a) 7.6e+09   b) 1.278123e-03   c) 5e-10

Exercise 2

1. a) LIST   b) RUN   c) DEL   d) END

2.
```
a) 15 PRINT 6*(5+4)    b) 10 PRINT 3^2    c) 10 LET A=6
   20 END                 20 END             20 LET Y=A*3
                             RUN             30 PRINT Y
                                             40 END
```

3. a) Yes   b) Yes   c) Yes   d) No   e) No   f) No

4. a) Yes   b) No   c) Yes   d) No   e) Yes   f) No

```
5. a) 15    b) 4   16
               7   49
               24
```

6. a) Add line number   b) At line 30 PRINT X*Y

c) Variable A must be given value before line 20

Exercise 3

1. Example output.

```
ENTER BASE & HEIGHT? 2,4
BASE   HEIGHT   AREA
 2       4       8
```

2. At line 20, change A;B to A,B

```
3. 10 PRINT"ENTER A VALUE FOR A";
   30 PRINT"ENTER A VALUE FOR B";
```

```
4. 20 INPUT A
   40 INPUT B
```

Exercise 4

1. a) line 50 b) line 90 c) line 70

   d) line 60 e) line 30 f) line 70

2. a) Both values of A and B will be printed, 19 and 5

   b) Only value of B will be printed, 10

   c) Only value of B will be printed, 20

```
3. a) 1    1         b) 5         c) 10
      3    27           3
      5    125          1
```

4. a) FOR X=1 TO 8   b) FOR X=0 TO 8 STEP 2

   c) FOR X=2.25 TO 18 STEP .25   d) FOR X=-5 TO 9 STEP 1.5

```
5. a) 10 FOR X=-10 TO 1  b) At line 10, delete , in front of STEP

      Output: 1              Output:  3
                                      5
                                      7
                                      .
                                      .
                                      .
                                      25
                                      729       5.19615243

  c) At line 40 NEXT C   d) At line 40 NEXT Y
                                    50 NEXT X
     Output:  4
              5              Output:  2
              6                       4
              7                       8
              8                       16
```

```
6. a) 10 FOR X=1 TO 10       b) 10 PRINT"WHAT IS YOUR NAME?"
      20 PRINT"(YOUR NAME)"     20 INPUT X$
      30 NEXT X                 30 FOR X=1 TO 10
      40 END                    40 PRINT X$
                                50 NEXT X
                                60 END
```

Exercise 5

1. a) SAVE   b) CATALOG   c) RENAME   d) DELETE

2. SAVE PROBLEM #1

3. DELETE EXERCISE #2

Student Exercises for Programming in BASIC - Math 20

The purpose of these programming exercises is to provide students with new and interesting ways to develop skill in applying mathematics. Using mathematics concepts and formulas in computer programs increases the student's familiarity and understanding of these concepts and formulas. In addition to this, the student can begin to understand the power of mathematical procedures when they are combined with a computer. And finally, the student becomes acquainted with the computer as a useful tool for processing mathematical data. None of these goals will be realized without considerable effort on your part. Good Luck!

The exercises given here are typical end-of chapter exercise whose solutions can be programmed in elementary BASIC. Programs may be written to solve questions containing specific numbers:

> Write a program to find the area of a triangle with base of 15 cm and height of 30 cm.

Or programs can be written so that any problem of that type may be solved using the program:

> Write a program to find the area of a triangle with a base(b) a height(h) to be specified by the program user.

A program written to solve a general problem such as the one above can be used to answer any number of specific examples of the given type. General programs are, however, more difficult to write.

Students are advised to start out writing a program to solve a specific example and then, later, revise the program to make it more general.

The exercises are arranged according to the Alberta Math 20 curriculum. In general, easier exercises are given first within a section. The teacher is reminded that there are usually several ways to program any solution. The most important criterion for a good program is whether it gives correct answers in some acceptable format. The student then is encouraged to verify his program by substituting in values which give known answers.

Of course, once you have a program it can be used to find the answer to any problem of that type. That is, on a computer any problem has only to be solved once. So do a good job of it.

Topic 1

Polynomials

1. Evaluate $g(x) = 5x^2 - 3x + 2$ at $g(3)$, $g(-2)$, $g(0)$.

2. If $h(x) = 2x^2 - 4x + 3$, evaluate $h(2)-h(-3)+h(1)$.

3. Find the zero of the following linear functions

   a) $y = 3x - 6$

   b) $y = ax + b$

4. Calculate approximate solutions for the following equations by trying different values for x. First use integer to find where the value of the polynomial changes sign and then narrow down this range.

   a) $x^2 - 6x + 3 = 0$

   b) $x^2 - x - 6 = 0$

   c) $ax^2 + bx + c = 0$

5. Given $P(x) = x^2 + 6x + 5$, determine if the polynomial has zeros for x values in the domain $-10 \leqslant x \leqslant 10$.

6. Make a table of ordered pairs of $y = 5x^2 - 3x + 2$ for values of x from -5 to 5 inclusive.

7. Write a program to

   a) derive a table of values using GOTO

   b) derive ordered pairs using FOR-NEXT

   c) derive ordered pairs using STEP

   d) derive ordered pairs for any quadratic function of the form

      $y = ax^2 + bx + c$

8. Factor $x^2 + 5x + 6$

Topic 2

Radicals

1. Evaluate $(\sqrt{2})^4$, $(\sqrt{3})^5$.

2. Write the following as single numbers.

   a) $\sqrt{16/25}$ b) $-\sqrt{144/9}$

   c) $\sqrt[3]{-125/64}$ d) $\sqrt[4]{81/625}$

3. Evaluate $y = 2 x$ for integral values of x from 1 to 10.

4. Given any mixed radical, write it as a complete radical.

   eg. $5\sqrt{2} = ?$, $2\sqrt{3} = ?$, $4\sqrt{5} = ?$

5. Evaluate a radical of the form

   a) $\sqrt[n]{R}$ b) $\sqrt[n]{R^m}$ or $(\sqrt[n]{R})^m$

Topic 3

Relations and Functions

1. For the function $y = 3x + 2$, $1 \leqslant x \leqslant 12$, $x \in N$ state the range.

2. For the function $y = x^2$, $-4 \leqslant x \leqslant 4$, $x \in I$ state the range.

3. List the y values for the equation $y = 3x^2 + 2$ given $-5 \leqslant x \leqslant 7$, $x \in I$.

4. List the ordered pairs for $y = -3x + 7$, where $-2.0 \leqslant x \leqslant -0.8$, x changing in steps of 0.2. Print the equation at the top and then list the ordered pairs in 2 columns titled

   X VALUES Y VALUES

5. Evaluate y in the equation $y = x^2 - 7x + 2$ for any x the user wants to input. Write the program so that the user will be given suitable instructions, and output that will be understandable.

Topic 4

Coordinate Geometry

The Straight Line

1. Find the slope of $3y - 2x + 3 = 0$.
2. Find the slope of the general linear function $Ax + By + C = 0$ and use it to determine the slope of any linear function in standard form ($Ax + By + C = 0$).
3. Given the two points (6,1) and (-2,3) determine the equation of the line passing through these points.
4. Find the x- and y-intercepts of $y = 3x + 7$.
5. Determine if the two equations are parallel.

   $y + 6x - 15 = 0$, $2x = 5 + y$.
6. Find the x- and y-intercepts of an equation of the form $Ax + By + C = 0$. If the equation does not have a particular intercept, a message should be displayed on the screen telling the user why. The user is to input the values of A, B, and C.
7. For a pair of straight lines described by the equations

   $Ax + By + C = 0$

   $Dx + Ey + F = 0$

   determine whether the lines are - parallel
   - perpendicular
   - neither of these.

   The user is to input the values of A, B, C, D, E, and F.
8. Write a program which will list suitable pairs which will enable you to sketch a graph of

   a) $y = 3x + 6$

   b) $y = mx + b$

Topic 5

Coordinate Geometry

1. Find the distance between (1,2) and (5,-2).
2. Find the distance between the point (A,B) and the point (C,D). The user is to input the values of A, B, C, and D.
3. Find the coordinates of the midpoint between the points (A,B) and (C,D). The user is to input A, B, C, and D.
4. Write a program for the slope between the points (-3,5) & (7,2).
5. Write a program to determine if the following three points are colinear: (1,2), (2,4), and (5,3).

Topic 6

System of Equations

1. Solve the following systems of equations by the method of substitution.

   $4x - 3y = 6, \quad y = 3$

2. Determine whether the following systems are independent, dependent, inconsistent.

   a) $y = 3x + 5, \ 2y = 6x + 5$

   b) $2x + y = -1, \ -4x + y = 11$

   c) $y = 2x + 1, \ y = x$

   d) $y = 3x + 5, \ 2y = 6x + 10$

3. Given two equations of the form

   $Ax + By + C = 0$

   $Dx + Ey + F = 0$

   determine if the system has a unique solution. The user is to input the values of A, B, C, D, E, and F.

Topic 7

Variation

1. How far apart are two people after 6 hours if they travel in the same direction at 10 km/hr and 15 km/hr?
2. The formula for blood pressure according to age is $P = 100+(1/2)A$ where A is age. Make a chart in 5 year intervals from 10 to 80 years.
3. Solving any problem involving direct variation when the general equation is in the form $y = cx$, where c is the constant.
4. Solving any problem involving inverse variation when the general equation is in the form $y = c/x$.

Topic 8

Geometry

The Circle

1. A point P is 25 cm. from the centre of a circle of radius 7 cm. Find the distance from point P to the point of tangency.
2. A circle has radius 5 cm. A chord is 6 cm. long. How far is the chord from the centre?
3. Given a central angle, determine the measure of the inscribed angle. Give the user suitable instructions and an understandable answer.
4. Finding the length of an arc given the measure of the sector angle and the radius.
5. Finding the area of a sector given the measure of the sector angle and the radius.

Topic 9

Trigonometry

1. Is a triangle of sides 3 cm, 10 cm, and 11.32 cm a right angle triangle?
2. Given Δ ABC with $A = 30^\circ$, $B = 60^\circ$, and $C = 90^\circ$ and $a = 1$, $b = 3$, $c = 2$, find $\sin 30^\circ$, $\cos 30^\circ$, $\tan 30^\circ$, $\sin 60^\circ$, $\cos 60^\circ$, and $\tan 60^\circ$.
3. What is the value of $\sin \theta$ if (3,8) lies on the terminal side of angle θ.
4. If cosine $\theta = 0.7$ find values of $\sin \theta$ and $\tan \theta$. Assume θ is between 0° and 90°.
5. Given an angle θ in degrees, write a program to convert it to radians.
6. Write a program that will display $\sin \theta$, $\cos \theta$, $\tan \theta$ if the angle is input in degrees.

Topic 10

Statistics

1. Place the following numbers in groups of width 1000.

   500, 2503, 3682, 1032, 5296, 10362, 9827, 8365, 5021, 5932, 6289, 8921, 7660, 4032.

   Find the frequency in each category.
2. Find the mean of any given series of numbers.
3. Find the median of any given set of 5 numbers. The user is to input the numbers in ascending or descending order, ie. sorting.
4. Determine Joe's percentile if on a test he was ranked 18th out of 60.

Topic 11

Quadratic Functions

1. Find the axis of symmetry for $y = 4x^2 + 4x + 5$.
2. Find the vertex, axis of symmetry, and the point of intersection with the y-axis of $y = 4x^2 + 4x + 5$.
3. Find the maximum or minimum value of $y = -3x^2 - 6x + 5$.
4. Given an equation of the form $Ax^2 + Bx + C = 0$, determine its roots. Check, and display an appropriate message on the screen telling the user whether the equation has
    - no real roots
    - one real root
    - two real roots.

   The user is to input A, B, and C.
5. Determine the range of $y = 4x^2 + 4x + 5$
6. Write a program to list suitable pairs which will enable you to sketch the graph of
    a) $y = x^2$
    b) $y = 5x^2$
    c) $y = (1/3)x^2$
    d) $y = 2x^2 + 3$
    e) $y = ax^2 + bx + c$
7. Evaluate the discriminant for
    a) $y = x^2 - x - 12$
    b) $y = 4x^2 - 4x + 1$
    c) $y = ax^2 + bx + c$

   (Have the computers print the characteristics of the roots. Real, non-real, etc.)

CURRICULUM AND INSTRUCTION GUIDE
FOR
COMPUTER PROGRAMMING IN MATHEMATICS

The Guide

This guide accompanies an elective which was developed for use in the Mathematics 20 course, a grade eleven mathematics course in the Alberta Provincial Curriculum. The elective, called Computer Programming in Mathematics(CPM) provides an experience in programming solutions to typical Math 20 exercises. It is designed to run parallel to the semestered course for approximately 15 one-hour-per-week sessions.The suggestions for teaching made in the guide are made on the assumption that the teacher is offering the CPM elective in this way. Quite possibly, the CPM program could be offered in different formats or even in conjunction with another mathematics course. In that case, the guidelines presented here would be modified.

The Goals of the CPM Elective

The advent of the microcomputer brings new challenges to mathematics teaching. The emphasis in the current CPM elective is on programming in BASIC. Students are given the opportunity to program, in BASIC, solutions to mathematics exercises and problems. Although the possibilities for using the microcomputer in mathematics teaching are numerous, the limited time available for the elective (15 hours) means that we keep a restricted focus. The elective amounts to a limited exposure to programming in BASIC (5 hours) plus programming assignments to students of typical Math

20 exercises. Before entering a serious discussion of the elective, we should stop to consider the benefits that students might gain from such an experience.

The most important benefit should be the deeper understanding gained as students use mathematical ideas in programming solutions. Part of the increased understanding should come simply from the practice students gain in using mathematics in their programs. But even further, because BASIC is a structured language, the students, once they are familiar with BASIC, should find their attention more directly focussed on the mathematical concepts involved. The focussing on the mathematics concepts will mean the student will learn the mathematics concepts more thoroughly. That is, because the format in which the questions are being answered is so straightforward and structured, the mathematical ideas that are being used are more available to be understood. In other words, the argument suggests that the programming experience is a very rich practice environment. This line of reasoning derives from the theory of information processing psychology.(*)This benefit will only be realized as students become very familiar with the language BASIC.

Although the above is the most important benefit, more direct benefits also exist. Programming in BASIC imposes an algorithmic format on the solutions of the mathematics exercises and, thereby promotes an algorithmic type of thinking by the student. Much of the learning in Math 20 is of an algorithmic nature. Practice in algorithmic thinking can directly contribute to the objectives of Math 20. In the above rationale, no claim is

* J. G. Greeno has written on this relating to mathematics.

made that algorithmic thinking is the main focus of the Math 20 course. It is one focus to which a programming experience can contribute. The student will, in the CPM elective, eventually write general programs where algorithms are dealt with at a general level, rather than just using an algorithm to solve a specific example. The student, in fact, in programming any solution, general or specific, is "teaching" the computer how to solve the exercise. After programming solutions to even thirty exercises, the student will have a substantial background in algorithmic thinking. In trying to realize this goal the teacher will be helping the student to achieve algorithmic competence and can provide direct help with student difficulties.

Another related benefit is the understanding students gain in some of the more fundamental ideas of the Math 20 course such as variable, literal coefficient, and function. The whole structure of BASIC centers around assigning values to variables. The LET command is a simply a means of defining a variable. Programming of general solutions to problems means understanding that the literal coefficients A, B, and C, for example, in the equation

$$Ax + By + C = 0$$

have a completely different role than x and y. The function idea is reinforced many ways in programming but especially through the idea of INPUTing values and PRINTing the output. The computer program in itself approximates a function. Although these ideas need much elaboration before they are fully understood, the teacher should definitely keep these benefits in mind when offering the CPM elective.

A benefit less directly related to the content of the Math 20 course is the step-by-step logical thinking that programming in BASIC promotes. The thinking that is done in working out BASIC programs in mathematics will almost directly carry over to regular classroom mathematics. At a lower cognitive level, the care with which a student works in programming should also carry over. Teachers offering the CPM elective encourage their students to be as careful in their regular classroom work as they are in their computer programs. Although general carry over of the type suggested in this paragraph is difficult to substantiate, teachers offering the CPM elective point to it as a distinct possibility.

A fourth benefit which should accrue is in students learning how computers can be used in mathematics. Students can gain an appreciation of how, once programmed, a computer can solve quickly and accurately many different mathematical problems and exercises. Seen in association with computers, mathematics appears to be a very powerful and useful subject. Such an experience could give students a whole new view of mathematics in modern technological society.

Although not related to the usual goals of Math 20, a final benefit to the student is that of learning some BASIC language, enough at least so they feel comfortable solving simple mathematics problems. The association of computers with mathematics is a highly realistic one and one from which both studies benefit.

Whether or not these goals are actually achieved is an open question. We may simply offer the elective because students like it and think it is important. Further work must be done to explore the possibilities mentioned here. But the teacher in offering the program should keep these goals firmly in mind and offer the elective to increase these benefits. The implementation of the CPM elective on which this document is based was monitored and has been reported separately in Integrating Programming into Mathematics. (*)The teacher planning to offer the CPM elective should examine the report carefully.

The CPM Elective

The elective consiscs of two parts i) instruction in the BASIC language and familiarity with the operation of a microcomputer and ii) student programming solutions to selected mathematics exercises. Because the elective is designed to be integrated into the Math 20 course, it is offered on a once-a-week basis in a semestered course. Instruction in BASIC and familiarization with the microcomputer is limited to five one-hour sessions. During the remaining ten sessions students are assigned exercises for programming. The topics for the programming exercises should parallel the topics of the course.

The actual structure of the program may vary according to particular school schedules. However there is usually some need within a school to regularly schedule the computer facility and the CPM elective benefits if students see the programming experiences reinforcing the work in the course. One caution is given: offering the elective less than once a week increases student forgetting and thereby increases the need for review. The description given in this manual is of a program offered one hour per week in a semestered course.

---

* Final Report, Alberta Education, 1984

One of the attractions of the computer is that it can be used in so many different ways in mathematics teaching. Because of this, the prospective teacher is warned of several things the CPM program is not intended to do. This is not a "course on programming" in BASIC. Only enough programming is taught to meet our specific purposes. It is not a course on how computers function, how they use binary numeration, nor how computers contribute to modern society. CPM does not encompass computer assisted instruction, either drill or tutorial. This warning is given, especially, because many educators feel the computer holds so much more for mathematics teaching. In sum, the CPM program is a short course in BASIC plus programming sessions based on typical exercises from the Math 20 course.

Five Lessons in BASIC

The Recommended Outline of Lessons (See page 2.) specifies the content to be covered in the first five lessons. Because these lessons are offered only once per week, each lesson begins with a review of the previous lesson. Students should be encouraged to read the Manual (See page 4.) and use it as a reference. Some teachers find success assigning students in groups to work through the Manual, especially if students who are fairly knowledgeable in programming are available to act as leaders. Whatever instructional approach is used, our findings suggest that unless the teacher is determined to stick to a tight schedule, they will not be able to cover the recommended material in the first five lessons. To this end we recommend following the topics proposed for each lesson quite closely.

The first three lessons

The first three lessons can be conducted in the classroom rather than in the computer lab. Having a microcomputer in the classroom for some demonstrations can be a help but is not essential. The first three lessons contain four key ideas:

i) components of a computer - such as the keyboard, disk drive
ii) immediate and deferred (program) mode
iii) arithmetic operations and program commands - such as LET, GOTO
iv) system commands - such as RUN, LIST

Although most of this learning is logical and makes sense in terms of the translation to the English language, the students are essentially learning to communicate in a new language(BASIC) to the computer. In this learning, while meaning is very important, a certain element of repetition is essential. This repetition is most effective when done verbally (orally). Repetition gained in writing programs on the computer is slow and cumbersome and therefore probably not very effective. During the first three lessons students will be anxious to get to the computer lab but in order to ensure they have a thorough grounding in the fundamentals of communicating with a computer, the teacher should consider delaying the visit until the fourth lesson.

The fourth lesson

Here the focus is on two very complex logical commands: IF-THEN and FOR-NEXT. These are probably best learned by writing and analyzing small programs which illustrate them. The first exposure to the computer lab inevitably leads to confusion for a class of thirty students. In general, instructions should be simple and explicit. The IF-THEN and FOR-NEXT commands can produce some exciting programs. The teacher should consider assigning knowledgeable student helpers to assist in the lab. Also students can be given programs which they simply have to enter into the computer and run.

The fifth lesson

The main goal of this lesson is the use of the disk and the disk drive. Initializing disks can cause problems. (See page 30.) Again instructions should be specific and foolproof. The idea is that a disk is a way of keeping a program for future reference. Names for programs should be systematic, specific and simple. This lesson becomes especially meaningful if the students have programs (even teacher-given ones) that they want to save. Our experience is that during the fourth and fifth lessons frustrations for the students can become extremely high, probably heightened by the wild expectations the students have for the computer. These frustrations will be an even greater concern if the teacher has to deal with more than one type of computer in the lab. This lesson is a pivotal one to the elective and so deserves special consideration.

Some variations

The instructions in the preceeding paragraph might seem overly strict. Several modifications are possible. The teacher for example may choose to begin the year with lessons on five consecutive days on BASIC. This format decreases the forgetting between lessons compared to lessons which are a week apart. Another advantage to this daily format is that, for various reasons, the computer lab is more available at the beginning of the year. Another possibility in the first five lessons is to introduce a "title page" program as an assignment.(*)This program does necessitate learning two new commands, VTAB and HTAB but the visual picture the program creates is highly motivating and errors made in programming show up dramatically. The program should be given to the student to type in. Some minor errors may be intentionally put in for the students to correct. The "title page" assignment should be given after the students can save programs on their disks where it can be used as a "Hello" program when initializing a disk.

After the first five lessons, the students should be ready to begin programming the Programming Exercises. In preparation for this, a teacher might want to introduce some of these exercises into the first five lessons. Simple exercises such as evaluating polynomials could be presented to advantage. Of course, the proof of the effectiveness of the first five lessons is in the ease with which the students take to programming the exercises in the following ones.

*See Appendix A on page 72 for the "title page" program

Programming Exercises - Lessons six to sixteen

The Math 20 curriculum includes eleven topics. Although not of equal length or importance, we can think of roughly one programming period for each topic. (See the Recommended Outline of Lessons on page 3.) Some topics such as analytic geometry and quadratic functions should be emphasized. The Programming Exercises section in the Handbook (See page 39.) gives several exercises under each topic. Until the teacher becomes familiar with the CPM program, some exercises from each topic should be assigned. Although the programming exercise ends when the program is written, the student can further profit by examining the results of the program. This is not normally emphasized in the CPM elective but the teacher should consider it.

Three different types of programming tasks

Within each topic, several types of exercises are possible. Each is appropriate for the CPM elective. Data generating involves evaluating expressions. Applying formulas involves using appropriate algorithms in given situations. More complex procedures involve putting two or more algorithms to use in the same program.

Data generating -

Here the program generates numerical data, such as ordered pairs for a given linear equation. Programming this is an end in itself. However, in addition, these ordered pairs can then be used to plot equations. Students could also look for patterns within the data or compare patterns of data from two linear equations. Once the program is written, the student can generate data for any set of equations which are of interest. Looking at patterns of numbers generated by given equations should lead to a better understanding of those equations. Without the availability of the computer, the task of generating this data is simply too great. The activity suggested here goes beyond what could be done in a regular non-computer mathematics lesson. This task most directly applies to the study of linear and quadratic functions.

Applying formulas -

Given a linear equation in the standard form, a formula for the slope of the equation exists. Knowing this formula, the student can write a program to determine the slope. Applying this program to two equations will allow the student to examine the two slopes and determine whether the graphs of the equations are parallel. The student can, in fact, apply this program to find the slopes of any number of equations. Again the computer allows for easy processing of innumerable examples. Patterns can be sought out. If the student combines comparison into one single program, we arrive at our third type of task: more complex procedures.

More complex procedures -

The computer can make decisions of less than, greater than and so on. This capability of making simple numerical decisions allows the programmer to write complex procedures where decisions must be made. This is particularly useful in many Math 20 exercises. For example, in solving a quadratic equation with the quadratic formula, the student would have the program first check on the value of the discriminant. If the discriminant is negative, a decision would probably be made to terminate the procedure. Here the student is building the decision-making process into the program.

For each of these tasks, the students write programs which generate answers. This in itself is worthwhile. However, with a little guidance, the student can then use the data in ways indicated above. Just as mathematical exercises in the regular program can be used in particular ways, computer generated answers hold even greater possibilities, simply because large numbers of them can be generated easily.

Three levels of specificity

The students' first attempts at programming any of the above tasks will centre around solving specific examples. However, with the use of INPUT, GOTO, and FOR-NEXT, programs can become more general. For example a program can be written to find the average of 2, 7, & 25 or it can be written to find the average of A, B, & C which may be specified by the user. For any of the above tasks, three levels of generality exist:

a) solving specific equations(procedures) with given numerical coefficients.

b) solving for a form of an equation(procedure), say the standard form of the linear equation, where any numerical coefficients may be INPUT.

c) solving for a pre-set range of the variables or numerical coefficient, using a FOR-NEXT loop.

These levels of generality do not especially represent levels of programming sophistication but rather levels of mathematical sophistication. Our general concern is to keep programming simple but the more mathematical sophistication the student can handle the better.

The teacher should consider the matter of assigning a series of exercises in a developmental fashion. The following series of exercises illustrates this idea:

1) Write a program to evaluate the expression $3x + 6$ at $x = 2,4,6$.
2) Write a program to evaluate the expression $3x + 6$ at any x which the user choses to specify.
3) Write a program to evaluate $y = 3x + 6$ and list the ordered pairs at all integral values of x between -20 and +20.
4) Write a program to evaluate $y = 3x + 6$ and list the ordered pairs at all values of x between two values which the user will specify.
5) The above exercise with the user having the option of the user specifying the steps in which x will proceed between the two values.

Admittedly the above sequence would be difficult for all Math 20 students to do but the teacher may make the first three compulsory with the last two being optional.

Assigning exercises

Depending on their difficulty two or three exercise can be assigned per weekly computer session. In general, students should be given the exercises beforehand. Ideally, programs should be written before the student comes to the keyboard to type them in. Most teachers find this works fairly well. In any case, students should begin to appreciate the value of thinking through a program before sitting down to the computer.

A second possibility in assigning exercises is to have the students use the program they have just written to solve ten exercises which may be taken from their mathematics textbook.(For an example see Appendix D page 75.) The first purpose this would serve has been mentioned above as "seeking out patterns." In this case the teacher would assign specifically chosen questions and probably would sequence them with some care. Another purpose for assigning several questions to be solved by the program would be that the student would see the point of writing "user-friendly" programs which could be easily "used." Finally using a program in this way (to work out ten assigned questions) would promote the idea of "checking" programs. Will the program work regardless of the value of the variable? What happens if a coefficient is zero? Our observations suggest that students must be shown the benefits of verifying their programs. Assigning mathematics exercises to be answered by a program is a good vehicle to do this.

A final consideration in making assignments concerns students revising previously written programs with the intent of making them more flexible, sophisticated or useable. From the point of view of computing theory, this is an attractive consideration. The programming exercises are not that complicated and student programming skills are not that sophisticated but clearly the same program could be combined with another program or written at a higher level of generality. Some teachers had success in encouraging students to revise their programs.

The programming exercises given in the Handbook should be used as a collection from which the teacher can select taking into consideration the points above. The actual number of exercises the students can program will vary from class to class and topic to topic. Initially the teacher should be careful not to overestimate the students' programming capability. Initial successful experiences will lessen the inevitable frustrations that students experience as they begin programming. Finally the assignments should be chosen in the light of an understanding of the goals of the CPM program.

General Instructional Considerations

Teachers who have taught the CPM elective have found at this stage of computer usage in schools that the average Math 20 class will have four or five students who have a some background in programming BASIC. As mentioned earlier, these students can help lead group discussions in initial classes, and in the computer lab they can be appointed helpers. Several cautions should be noted. First the student-helpers may simply write programs for students and not elaborate on their programming procedures. The helpers also tend to write programs that are too sophisticated for our purposes. Of course the ultimate in programming, like mathematics, is elegance. However, from our point of view the more deliberate and explicit the program, the more obvious is the mathematics concept. Until a student fully understands BASIC as a language, programs should be kept simple. Excessive use of student-helpers might also detract from the helpers' experiences in the CPM program. The teacher is urged to give sincere consideration to using to the best advantage any previous computer background of students. Even these students find the CPM elective an interesting and profitable challenge.

Another general concern of CPM teachers is how much direct help to give students in writing programs. Initially students have a hard time to grasp what it means to write a program. At the extreme, some students have begun by typing onto the screen the question:"What is the slope?" They are then surprised that the computer doesn't respond. Although this is an extreme example, students have very fuzzy ideas of what input is required and what output can be expected. Teachers have usually found it necessary to give students explicit examples of programs. The question that arises is:"How explicit and how many?" before students can program on their own. Although the question is not easily answered some comments can be made. Giving students programs with errors which they have to correct seems to be useful. In this way they get the sense of the structure of the program while still having to fill in details. Another way is to give examples with two or three crucial lines and ask them to begin and end the program. The answer to the question of how many direct examples to give lies in the quality of the examples and the possibilities of them being used in instructive ways. Finally the teacher should realize that example programs can be not only run on the computer but also used as a basis for class discussion where students explain the function of various lines.

Another general instructional consideration is whether teachers should insist that the programs written should be "user-friendly." This is a computer term which means that any user can easily use the program. Clearly the main user of the program is the student. Therefore the first guideline should insure that even two or three weeks after a student has saved a program on a disk, the program can be used easily. Programs should also be written in standard English. Just as teachers expect hand written solutions to mathematics problems to be self explanatory, so should programs written on the computer be. In a sense, the benefit the student derives from answering exercises of any kind is somewhat proportional to the quality of the response to the exercise. Writing "user-friendly" programs should be interpreted to mean the same as well-written, self-explaining, high-quality answers. The above guidelines would not mean that text needs to be centred on the screen, certainly not in flashing letters. If students have routines to "personalize" programs(that is, enter the user's name) or to ask the user if he/she wants to continue, they should use them provided the routines do not detract from their programming experiences.

Teachers of the CPM program have also generally found four or five students in their classes who need significantly more help than others. The teacher should be aware that such students exist and be advised to monitor their progress closely. In many cases these students seem not to be aware of what is expected of them. Pair them with students who can help them. Require minimal programs at first. Our experience suggests that all Math 20 students can relate to the CPM elective and benefit.

Enrichment

One of the fears of computing teachers in schools is of students who may know more about computers than they do. This is quite common in the CPM elective in Math 20. Although teachers' fears are important, a more relevant question is whether the CPM elective is redundant for these students. In fact the more familiarity the student has with BASIC the better the chances of his truly benefiting in a mathematical sense from the elective. Enrichment possibilities within CPM are great especially because the programming component of the CPM program is an individualized experience. More knowledgeable students write more sophisticated programs. There seems to be no upward limit to what students can do on their own.

Specific enrichment activities also exist. Students could look into programming graphics. They could, for example, be given a "plotting program" which plots graphs of linear or quadratic functions(See Appendix B page 73.) and be asked to modify it and/or demonstrate it to the rest of the class. Mathematics teaching journals often contain programming suggestions such as generating triangular numbers. The National Council of Teachers of Mathematics has, for example, a disk full of simple programs in mathematics.(*) Teachers have found, however, that they do not have to resort to extra assignments for enrichment. In general,

* NCTM: Mathematics Disk, 1984

students with a good background in computing are more often a blessing than a problem.

CPM As An Elective

Before the CPM as an elective portion of the Math 20 program can be offered a class-size number of microcomputers must be available for one hour per week. Of course other time schedules are possible. Because the elective is so demanding, teachers have found it impossible to run the during noon hours. By far the best arrangement is for the class to be scheduled into the computer lab in a regular mathematics period. Each student should have access to a computer for the full period. A computer lab with two or more different types of computers creates endless frustrations. Since the focus of the CPM elective is on using the computer as a tool with minimal instructions, explicit efficient instruction must be given in its operation. This is almost impossible with more than one type of computer. If the computer lab has more than one type of computer, the teacher must become totally familiar with all aspects of operation of each type. Even then, helping 30 students in a lab situation can become chaotic.

Grading

Because the CPM elective is a part of the Math 20 program, grading is an essential part of the experience. The computer's capability to copy programs adds a problematic dimension to grading. Teachers should be aware of this possibility and make the conditions of acceptability clear to the students. CPM teachers reported no difficulty in dealing with this matter. Teachers found that grading all of the students' programs was impossible. Because the students, themselves, can easily check whether a program works, a sensible approach is to ask students to submit one program per topic for grading. Beyond this students could submit programs of special interest for which bonus marks can be given. The elective should be treated as equivalent to any unit in the course and for grading purposes weighted accordingly.

Four guidelines for determining the elective grade are suggested in the grading sheet.(See Appendix C page 74.) They are

1. participation in the course
2. compulsory programs
3. interest, creativity and attitude shown by the student
4. amount of programming done beyond the minimum

A hardware requirement in connection with grading is for teachers to have access to a computer during their preparation period. Grading is a matter to which the teacher should give careful attention. At best it is time consuming.

Relation to the Mathematics Curriculum

The CPM elective was offered on a one hour per week basis. This hour was usually spent in the computer laboratory. The elective in this way took 15 hours out of the regular Math 20 course. In addition to these 15 hours, some teachers found it necessary to assign computer exercises in the class hour previous to the computer laboratory hour. Teachers in going over certain lessons would alert students to the potentialities for programming solutions to certain types of exercises. This was especially possible because usually the computer activities were based on content being taken in the class at that time. Although a teacher should be watchful of the CPM elective taking extra time away from the regular program, a certain amount of overlap is healthy.

Besides this intentional mention of computer activities in the regular mathematics program, many incidental references were made. Teacher were observed to use "LET", a BASIC command, in an explicit way in their teaching. Obviously the "let" statement in mathematics is used differently than in BASIC but it serves a similar function of identifying variables. Teachers were, in this way, assisting in the transfer of computer-type thinking to mathematics. Teachers often encouraged students to write the solutions to their regular classroom mathematics problems as carefully as they had to write their programmed solutions. This again is an attempt by the teacher to assist in the transfer of computer developed habits to mathematics. Teachers should try to become aware of possible areas of overlap that they can usefully exploit. Clearly the programming activity will have the greatest impact on the mathematics curriculum when appropriate transfer occurs.

Incidentally, teachers felt that the amount of mathematics being done in the computer periods increased as the student became more competent in programming in BASIC. Students in the CPM elective should be encouraged to use their time to do more mathematics NOT to increase their competence as programmers in accordance with the goals of CPM.

Student Reaction

The most important aspect of the CPM elective is the challenge that it presents for most students and therefore for most teachers. Given the current high profile of computers in society, students probably enter the elective with rather too high expectations. Teachers approaching the elective naively will find frustration levels building dramatically over the first four or five lessons peaking the first time students are alone at the computer. In spite of this, students are very positive about the experience. They apparently believe that the experience is good for them. The teacher should use this positive attitude to the best benefit by striving to keep it high. Although computers in general seem to attract more boys than girls, girls in the CPM elective appear to do equally well.

Teacher Preparation

In order to offer the CPM elective no extensive teacher preparation is necessary. The teacher should have familiarity with BASIC to the level of the STUDENT MANUAL. Of course the teacher should expect to extend his knowledge of programming as the course proceeds. But equally important to a knowledge of BASIC is complete familiarity with the operational aspects of the particular computer installation in the school. This last advice is doubly important if the computer lab contains more than one type of computer. Idiosyncrasies of particular computers and installations are frustrating at any time but with a roomful of thirty anxious students, it is murder. The teacher who is less knowledgeable must develop ways of employing the inevitable "computer hot-shots" in all Math 20 classes. Following these few guidelines, no Math 20 teacher need shy away from offering the CPM elective.

Teacher Reaction

Teachers who offered the elective over two semesters have reacted positively to the experience. The justification for the elective lies in the contribution it makes to understanding the mathematics of the Math 20 course and of the understanding of the computer's relation to mathematics. In fact, if students had a knowledge of BASIC the computer experience could simply be offered as part of the regular program, not an elective at all. Someday, when all students come to grade eleven with a knowledge of BASIC this will be possible.

As is true in any curriculum implementation, teachers found that they were re-thinking the curriculum. For example, the emphasis on the literal coefficient in linear and quadratic equations prompted teachers to re-examine their approach to the topic. Teachers also tended to pay more attention to laying-out answers to problems more explicitly. The idea that a program could be verified by specifying certain values of variables brought back the idea of verifying any mathematical solution to a problem. The whole matter of defining variables, especially through the use of "let", was seen in a new light. The programming of solutions to mathematics exercises encouraged teachers to re-think their own approach to mathematics and to teaching it.

Lastly, teachers felt that the students through writing computer solutions were reviewing their mathematics in a high quality manner. Not simply rotely taking formulas and substituting in numbers but thinking through the formulas and using them in a larger perspective. In all of the computer programming exercises, the students were reinforcing the mathematics of the classroom. Finally, students found the experience to be worthwhile; knowing this, teachers will be encouraged to offer, improve and extend this fascinating new dimension of the mathematics curriculum.

The Future of the CPM ELective

The CPM elective will continue to be offered only if teachers become convinced of its benefits. A report, Integrating Programming into Mathematics (*) of an attempt to monitor its implementation is available. Other efforts to study the CPM program should be made. Regardless of the experimental studies made, teachers are encouraged to think about the program as they offer it. Only through offering the program can it be developed, extended and improved. Like most significant educational changes, the CPM is a complex educational activity, the fact that it must integrate with the Math 20 course makes its implemenation even more challenging. The five teachers who offered the elective over two different semesters see its benefit as multi-dimensional. Perhaps the major benefit is that the course allows students to see mathematics in a new light, as a useful powerful tool in a modern technological society.

* Final Report, Alberta Education, 1984

Appendix A

Title Page

```
 10 HOME
 20 REM * A DRESSED-UP HELLO PROGRAM *
 30 VTAB 5:HTAB 7
 40 REM * TOP OF BORDER *
 50 PRINT"***...26 times"
 60 REM * LOOPS FOR SIDES OF BORDER *
 70 FOR I=6 TO 18
 80 HTAB 7:PRINT"*":NEXT I
 90 FOR I=6 TO 18
100 HTAB 32:VTAB I:PRINT"*":NEXT I
110 REM * BOTTOM OF BORDER *
120 VTAB 19:HTAB 7
130 PRINT"***...26times"
140 REM * INSIDE OF BORDER *
150 VTAB 8:HTAB 16:PRINT"MATH 20"
160 VTAB 10:HTAB 13:PRINT"COMPUTING UNIT"
170 VTAB 14:HTAB 13:PRINT"SUSIE SUMMER"
180 VTAB 16:HTAB 11:PRINT"SEMESTER I-1984"
190 VTAB 22:HTAB 8:INVERSE
200 PRINT"PRESS C TO CONTINUE"
210 GET W$
220 IF W$<>"C" THEN 210:NORMAL
230 END
```

Appendix B

Plotting Program

```
 10 TEXT:HOME
 20 VTAB 12
 30 PRINT"THIS PROGRAM GRAPHS LINES"
 40 PRINT"GIVEN IN THE GENERAL FORM"
 50 PRINT"Y=M*X+B."
 60 GOSUB 1000
 70 HOME
 80 HGR
 90 HCOLOR=2
100 PRINT:POKE 37,20:PRINT
110 PRINT"FORM Y=M*X+B"
120 PRINT"INPUT M AND B"
130 INPUT M,B
140 PRINT:PRINT:PRINT
150 REM DRAW AXES
160 HPLOT 140,0 TO 140, 159
170 HPLOT 0,80 TO 279,80
180 REM MARK AXES
190 FOR H=0 TO 270 STEP 10
200 HPLOT H,77 TO H,83
210 NEXT
220 FOR K=0 TO 160 STEP 10
230 HPLOT 137,K TO 143,K
240 NEXT
250 REM MAKE LABELS
260 HPLOT 271,65 TO 277,71
270 HPLOT 277,65 TO 271,71
280 HPLOT 147,0 TO 151,6
290 HPLOT 155,0 TO 147,12
300 REM PLOT THE STRAIGHT LINE
310 FOR X=-14 TO 13.9 STEP .1
320 Y = M*X + B
330 IF Y > 8 THEN 340
340 IF Y < -7.9 THEN 340
350 HPLOT 140+10*X,80-10*Y
360 NEXT X
370 REM GIVE THE EQUATION
380 PRINT:POKE 37,20:PRINT
390 PRINT"THIS IS THE GRAPH OF Y=";M;" X+";B
400 PRINT:PRINT"ANOTHER, TYPE 1;QUIT, TYPE 0"
410 INPUT Z
420 PRINT:IF Z=0 THEN 2000
430 PRINT"DO YOU WISH THE NEXT GRAPH"
440 PRINT"ON THE SAME AXES? YES OR NO?"
450 INPUT R$
460 IF R$ = "YES" THEN 90
470 IF R$ = "NO" THEN 60
480 GOTO 430
490 VTAB 24
500 PRINT"TYPE C TO CONTINUE"
510 GET C$:IF C$ <> "C" THEN 510
520 RETURN
530 TEXT:HOME:PRINT"BYE FOR NOW."
```

Appendix C

Typical Grading Schedule

MATH 20 COMPUTING UNIT

NAME________________________

Your mark in this unit is determined by the following standard:

1. Participation in the program ______

(20)

2. Programs written for

   1. Radicals ______
   2. Polynomials ______
   3. Relations and functions ______
   4. Coordinate Geometry ______
   5. Systems of Equations ______
   6. Quadratic Functions ______
   7. Variations ______
   8. Geometry and Trigonometry ______
   9. Statistics ______

______

(20)

3. Interest, creativity and attitude ______

(10)

4. Number of programs beyond minimum ______

(10)

Appendix D

Typical 10 Question Worksheet

For each of the following quadratic functions, determine the following: (Use your computer programs.)

1) the coordinate of the vertex
2) the x-intercepts
3) the y-intercept
4) the equation of the axis of symmetry
5) the max./min. value
6) the range

a) $y = 4x^2 - 4x - 1$

b) $y = -x^2 + 10x - 16$

c) $y = 3x^2 + 6x - 9$

d) $y = 3x^2 - 6x + 15$

e) $f(x) = x^2 - 1$

f) $f(x) = -x^2 - 2x + 2$

g) $y = (1/3)x^2 + 2x - 5$

h) $y = (2x-3)(2x+5)$

i) $y = 2x^2 - 360x + 16102$

j) $y = 4x^2 - 4ax + a - b$

Selected References

Mathematics Texts:

Fryer, K.D., Dunkley, R.G., Elliott, H.A., Hill, N.J., and MacKay, R.J., Holt Mathematics 5. Waterloo, Ont.: Holt, Rinehart and Winston of Canada, Ltd., 1980.

Kelly, B., Alexander, B., and Atkinson, P., Mathematics 10 - Addison-Wesley Secondary Mathematics. Don Mills, Ont.: Addison and Wesley Publisher 1982.

Yunker, L.E., Vannatta, G.D., and Crosswhite, F.J., Merrill-Advanced Mathematical Concepts. Columbus, Ohio: Bell & Howell Company, 1981.

General Programming Books:

Cathcart, G.M., and Cathcart, W.G., Programming Exercises in BASIC for Microcomputers. Toronto, Ont.: Gage Publishing Ltd., 1984.

Mathematics Problems for Programming:

Kieren, T.E., CAMP Intermediate Mathematics. Atlanta: Scott, Foresmann and Company, 1970.

Spencer, D.D., Problems for Computer Solution. 2nd ed. Rochelle Park, N. J.: Hayden Book Co., 1977.

SOLUTION PROGRAMS FOR STUDENT EXERCISES IN PROGRAMMING

Topic 1

Polynomials

Problem 1

```
10 REM EVALUATE G(X) = 5X^2-3X+2
20 PRINT"ENTER A VALUE FOR X";
30 INPUT X
40 LET Y=5*(X^2)-(3*X)+2
50 PRINT"X = "X", G("X") = "Y
60 GOTO 10
70 END
```

Problem 2

```
10 REM EVALUATE H(X) = 2X^2-4X+3
20 FOR I=1 TO 3
30 PRINT"ENTER #"I" VALUE OF X";
40 INPUT X(I)
50 LET H(I)=2*(X(I)^2)-(4*X(I))+3
60 NEXT I
70 FOR I=1 TO 3
80 PRINT"H("X(I)") = "H(I)
90 NEXT I
100 LET Y=H(1)-H(2)+H(3)
110 PRINT"H("X(1)")-H("X(2)")+H("X(3)") = "Y
120 END
```

Problem 3

```
10 REM FIND THE ZERO OF FUNCTION Y = AX+B
20 PRINT"ENTER VALUES FOR A, B";
30 INPUT A,B
40 LET S=-B/A
50 PRINT"Y = 0 WHEN X = "S
60 GOTO 10
70 END
```

Problem 4

```
10 REM APPROXIMATE SOLUTIONS FOR A(X^2)+BX+C = 0
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 PRINT"Y = "A"(X^2)+"B"X+"C
40 INPUT"ENTER A VALUE FOR X";X
50 LET Y=A*(X^2)+B*X+C
60 PRINT"X = "X": Y = "Y
70 GOTO 40
80 END
```

Problem 5

```
10 PRINT"P(X) = X^2+6X+5"
20 FOR X=-10 TO 10
30 LET P=(X^2)+(6*X)+5
40 IF P=0 THEN PRINT"X = "X,"P = "P
50 NEXT X
60 END
```

Problem 6

```
10 PRINT"Y = 5(X^2)-3X+2, -5<=X<=5"
20 PRINT"(X , Y)"
30 FOR X=-5 TO 5
40 LET Y=5*(X^2)-(3*X)+2
50 PRINT"("X","Y")"
60 NEXT X
70 END
```

Problem 8

```
10 PRINT"FACTOR POLYNOMIAL OF THE FORM (X^2)+BX+C"
20 INPUT"ENTER VALUES FOR B,C";B,C
30 IF C<0 THEN 70
40 IF B<0 THEN 60
50 FOR M=-B TO B:GOTO 80
60 FOR M=B TO -B:GOTO 80
70 FOR M=C TO -C
80 IF M*(B-M)<>C THEN 100
90 PRINT"X^2+"B"X+"C" = (X+"M")(X+"B-M")"
100 NEXT M:GOTO 10
110 END
```

Topic 2

Radicals

Problem 1

```
10 PRINT"EVALUATE RADICAL OF THE FORM (A^(1/N))^C"
20 INPUT"ENTER A VALUE OF RADICAND";A
30 INPUT"ENTER AN INDEX OF RADICAL";N
40 INPUT"ENTER THE EXPONENT NUMBER";C
50 LET Y=(A^(1/N))^C
60 PRINT"("A"^(1/"N"))^"C" = "Y
70 GOTO 10
80 END
```

Problem 2

```
10 PRINT"EVALUATE RADICAL OF THE FORM A^(1/N)"
20 INPUT"ENTER A VALUE OF RADICAND";A
30 INPUT"ENTER AN INDEX OF RADICAL";N
40 LET Y=A^(1/N)
50 PRINT A"^(1/"N") = "Y
60 GOTO 10
70 END
```

Problem 3

```
10 PRINT"EVALUATE RADICAL OF THE FORM Y = 2(X^.5)"
20 FOR X=1 TO 10
30 LET Y=2*(X^.5)
40 PRINT"X = "X,"Y = "Y
50 NEXT X
60 END
```

Problem 4

```
10 REM WRITE ANY MIXED RADICAL AS A COMPLETE RADICAL
20 INPUT"ENTER A POSITIVE REAL NUMBER";A
30 INPUT"ENTER A VALUE OF RADICAND";B
40 INPUT"ENTER AN INDEX OF RADICAL";N
50 LET Y=(A^N)*B
60 PRINT A"*("B"^(1/"N")) = "Y"^(1/"N")"
70 GOTO 10
80 END
```

Topic 3

Relations and Functions

Problem 1

```
10 PRINT"EVALUATE Y=3X+2, 1<=X<=12"
20 FOR X=1 TO 12
30 LET Y=(3*X)+2
40 PRINT"X = "X,"Y = "Y
50 NEXT X
60 END
```

Problem 2

```
10 PRINT"EVALUATE Y=X^2, -4<=X<=4"
20 FOR X=-4 TO 4
30 LET Y=X^2
40 PRINT"X = "X,"Y = "Y
50 NEXT X
60 END
```

Problem 3

```
10 PRINT"LIST THE Y VALUES OF Y=3(X^2)+2, -5<=X<=7"
20 FOR X=-5 TO 7
30 LET Y=3*(X^2)+2
40 PRINT"Y = "Y
50 NEXT X
60 END
```

Problem 4

```
10 PRINT"LIST THE ORDERED PAIRS FOR Y=-3X+7, -2.0<=X<=-0.8
20 PRINT"X VALUES","Y VALUES"
30 FOR X=-2.0 TO -0.8 STEP 0.2
40 LET Y=(-3*X)+7
50 PRINT X,Y
60 NEXT X
70 END
```

Problem 5

```
10 PRINT"ENTER A VALUE OF X FOR Y=(X^2)-7X+2"
20 INPUT X
30 LET Y=(X^2)-(7*X)+2
40 PRINT"THE ANSWER IS Y = "Y
50 GOTO 10
60 END
```

Topic 4

Coordinate Geometry

(The Straight Line)

Problem 1

```
10 REM FIND SLOPE OF 3Y-2X+3 = 0
20 LET A=-2
30 LET B=3
40 PRINT"SLOPE = ";-A;"/";B
50 END
```

Problem 2

```
10 REM FIND SLOPE OF AX+BY+C = 0
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 PRINT"LINEAR FUNCTION: "A"X+"B"Y+"C" = 0"
40 IF B=0 THEN PRINT"SLOPE IS UNDEFINED.":GOTO 20
50 PRINT"SLOPE = "(-1)*A"/"B
60 GOTO 20
70 END
```

Problem 3

```
10 REM TWO POINTS EQUATION
20 LET X1=6:Y1=1
30 LET X2=-2:Y2=3
40 PRINT"THE LINE IS Y-"Y1" = "(Y2-Y1)"/"(X2-X1)"(X-"X1")"
50 END
```

Problem 4

```
10 REM FIND X- AND Y-INTERCEPT OF Y=3X+7
20 LET A=3
30 LET C=7
40 PRINT"X-INTERCEPT IS ("(-1*C)/A",0)"
50 PRINT"Y-INTERCEPT IS (0,"C")"
60 END
```

Problem 5

```
10 PRINT"DETERMINE IF Y+6X-15=0, 2X=5+Y ARE PARALLEL"
20 LET A=6:B=1
30 LET D=2:E=-1
40 LET M1=-A/B
50 LET M2=-D/E
60 PRINT"M1 = "M1,"M2 = "M2
70 IF M1=M2 THEN PRINT"LINES ARE PARALLEL":END
80 PRINT"LINES ARE NOT PARALLEL"
90 END
```

Problem 6

```
10 PRINT"FIND X- AND Y-INTERCEPTS OF AX+BY+C = 0"
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 IF A=0 THEN PRINT"NO X-INTERCEPT. LINE // TO X-AXIS":
                                                 GOTO 10
40 IF B=0 THEN PRINT"NO Y-INTERCEPT. LINE // TO Y-AXIS":
                                                 GOTO 10
50 LET M=-C/A
60 LET N=-C/B
70 PRINT"X-INTERCEPT IS ("M",0)"
80 PRINT"Y-INTERCEPT IS (0,"N")"
90 GOTO 10
100 END
```

Problem 7

```
10 PRINT"DETERMINE AX+BY+C=0 & DX+EY+F=0 ARE PARALLEL"
20 PRINT"OR PERPENDICULAR"
30 INPUT"ENTER VALUES FOR A, B, C";A,B,C
40 INPUT"ENTER VALUES FOR D, E, F";D,E,F
50 IF B=0 AND E=0 THEN PRINT"SLOPES OF BOTH LINES ARE
                                     UNDEFINED":GOTO 10
60 IF B=0 THEN PRINT"SLOPE OF 1ST LINE IS UNDEFINED":GOTO 10
70 IF E=0 THEN PRINT"SLOPE OF 2ND LINE IS UNDEFINED":GOTO 10
80 LET M1=-A/B: M2=-D/E
90 IF M1=M2 THEN PRINT"LINES ARE PARALLEL":GOTO 10
100 IF M1*M2=-1 THEN PRINT"LINES ARE PERPENDICULAR":GOTO 10
110 PRINT"LINES ARE NEITHER PARALLEL NOR PERPENDICULAR":
                                                GOTO 10
120 END
```

Topic 5

Coordinate Geometry

Problem 1

```
10 PRINT"FIND THE DISTANCE BETWEEN (1,2) AND (5,-2)"
20 LET X1=1:Y1=2
30 LET X2=5:y2=-2
40 LET D=((X2-X1)^2+(Y2-Y1)^2)^.5
50 PRINT"THE DISTANCE = "D
60 END
```

Problem 2

```
10 PRINT"FIND DISTANCE BETWEEN POINTS (A,B) AND (C,D)"
20 INPUT"ENTER A VALUE FOR A";A
30 INPUT"ENTER A VALUE FOR B";B
40 INPUT"ENTER A VALUE FOR C";C
50 INPUT"ENTER A VALUE FOR D";D
60 LET X=(C-D)^2
70 LET Y=(D-B)^2
80 LET Z=(X+Y)^.5
90 PRINT"THE DISTANCE BETWEEN THE POINTS = ";Z
100 GOTO 10
110 END
```

Problem 3

```
10 PRINT"FIND THE MIDPOINT OF (A,B) AND (C,D)"
20 INPUT"WHAT IS THE VALUE OF A";A
30 INPUT"WHAT IS THE VALUE OF B";B
40 INPUT"WHAT IS THE VALUE OF C";C
50 INPUT"WHAT IS THE VALUE OF D";D
60 LET X=(A+C)/2
70 LET Y=(B+D)/2
80 PRINT"THE MIDPOINT IS ("X","Y")"
90 GOTO 10
100 END
```

Topic 6

System of Equations

Problem 1

```
10 REM * SOLVE SYSTEM OF EQUATIONS BY SUBSTITUTION
20 LET Y=3
30 LET X=(3*Y+6)/4
40 PRINT"4X-3Y = 6, Y = 3"
50 PRINT"X = "X
60 END
```

Problem 2

```
10 REM DETERMINE WHETHER LINEAR SYSTEM OF THE FORM AX+BY=C
20 REM AND PX+QY=R ARE CONSISTENT, INCONSISTENT, DEPENDENT
30 REM OR INDEPENDENT
40 INPUT"ENTER VALUES FOR A, B, C";A,B,C
50 INPUT"ENTER VALUES FOR P, Q, R";P,Q,R
60 IF A<>0 AND B<>0 AND C<>0 AND P<>0 AND Q<>0 AND R<>0
                                                     THEN 160
70 IF A<>0 AND B<>0 AND C=0 AND P<>0 AND Q<>0 AND R=0
                                                     THEN 190
80 IF A<>0 AND B<>0 AND C=0 AND P<>0 AND Q<>0 AND R<>0
                                                     THEN 170
90 IF A<>0 AND B<>0 AND C<>0 AND P<>0 AND Q<>0 AND R=0
                                                     THEN 200
100 IF A<>0 AND B=0 AND C<>0 AND P<>0 AND Q=0 AND R<>0
                                                     THEN 220
110 IF A=0 AND B<>0 AND C<>0 AND P=0 AND Q<>0 AND R<>0
                                                     THEN 240
120 IF A<>0 AND B=0 AND C=0 AND P<>0 AND Q=0 AND R=0 THEN 260
130 IF A=0 AND B<>0 AND C=0 AND P=0 AND Q<>0 AND R=0 THEN 260
140 IF A<>0 AND B=0 AND C<>0 OR C=0 AND P=0 AND Q<>0 AND
                                          R<>0 OR R=0 THEN 280
150 IF A=0 AND B<>0 AND C<>0 OR C=0 AND P<>0 AND Q=0 AND
                                          R<>0 OR R=0 THEN 280
160 IF A/P=B/Q AND B/Q=C/R THEN 260
170 IF A/P=B/Q AND B/Q<>C/R THEN 270
180 IF A/P<>B/Q THEN 280
190 IF A/P=B/Q THEN 260
200 IF P/A=Q/B AND Q/B<>R/C AND R/C=0 THEN 270
210 IF P/A<>Q/B THEN 280
220 IF A/P=C/R THEN 260
230 IF A/P<>C/R THEN 270
240 IF B/Q=C/R THEN 260
250 IF B/Q<>C/R THEN 270
260 PRINT"THE SYSTEM IS CONSISTENT AND DEPENDENT":END
270 PRINT"THE SYSTEM IS INCONSISTENT":END
280 PRINT"THE SYSTEM IS CONSISTENT AND INDEPENDENT":END
```

Problem 3

```
10 PRINT"DETERMINE IF AX+BY+C = 0 AND DX+EY+F = 0"
20 PRINT"HAVE A UNIQUE SOLUTION"
30 INPUT"ENTER VALUES FOR A, B, C";A,B,C
40 INPUT"ENTER VALUES FOR D, E, F";D,E,F
50 IF A/D=B/E AND B/E<>C/F THEN PRINT"THE SYSTEM HAS NO
                                  SOLUTIONS":GOTO 10
60 IF A/D<>B/E THEN PRINT"THE SYSTEM HAS A UNIQUE
                                   SOLUTION":GOTO 10
70 PRINT"THE SYSTEM HAS AN INFINITE NUMBERS OF SOLUTIONS":
                                              GOTO 10
80 END
```

Topic 7

Variations

Problem 1

```
10 REM DISTANCE BETWEEN TWO PEOPLE AFTER 6 HOURS
20 LET D1=6*10
30 LET D2=6*15
40 PRINT"AFTER 6 HRS PERSON A TRAVELED "D1" KMS."
50 PRINT"AFTER 6 HRS PERSON B TRAVELED "D2" KMS."
60 PRINT"DISTANCE BETWEEN 2 PERSONS IS "D2-D1" KMS."
70 END
```

Problem 2

```
10 PRINT"BLOOD PRESSURE FORMULAR IS P = 100+(1/2)A"
20 PRINT"   AGE","CLASS AGE","BLOOD PRESSURE"
30 FOR A=10 TO 75 STEP 5
40 LET P=100+(1/2)*(2*A+5)/2
50 PRINT A"-"A+5,(2*A+5)/2,P
60 NEXT A
70 END
```

Problem 3

```
10 PRINT"S^2 VARIES DIRECTLY AS T^3, S=2 WHEN T=4"
20 PRINT"FIND S WHEN T=8, FIND T WHEN S=16"
30 PRINT"S^2 = C(T^3) OR C = (S^2)/(T^3)"
40 LET S=2: T=4
50 LET C=(S^2)/(T^3)
60 PRINT"C = "C
70 LET T=8: S=(C*(T^3))^.5
80 PRINT"WHEN T=8, S="S
90 LET S=16: T=((S^2)/C)^(1/3)
100 PRINT"WHEN S=16, T="T
110 END
```

Problem 4

```
10 PRINT"X VARIES INVERSELY AS Y^2, X=3.5 WHEN Y=5"
20 PRINT"CALCULATE VALUE  OF Y WHEN X=14"
30 PRINT"X = C/(Y^2) OR C = X(Y^2)"
40 LET X=3.5: Y=5
50 LET C=X*(Y^2)
60 PRINT"C = "C
70 PRINT"Y = (C/X)^.5"
80 LET X=14
90 LET Y=(C/X)^.5
100 PRINT"WHEN X = 14, Y = "Y
110 END
```

Topic 8

Geometry

(The Circle)

Problem 1

```
10 PRINT"FIND DISTANCE FROM POINT P TO POINT OF TANGENCY"
20 LET A=25: B=7
30 LET C=(A^2-B^2)^.5
40 PRINT"DISTANCE FROM P TO POINT OF TANGENCY = "C" CM."
50 END
```

Problem 2

```
10 REM * DISTANCE FROM CHORD TO CENTRE OF A CIRCLE
20 LET A=5: B=3
30 LET C=(A^2-B^2)^.5
40 PRINT"DISTANCE FROM CHORD TO CENTRE OF CIRCLE = "C" CM"
50 END
```

Problem 3

```
10 PRINT"GIVE ME A CENTRAL ANGLE IN DEGREE";
20 INPUT C
30 LET B=C/2
40 PRINT"THE INSCRIBED ANGLE = "B" DEGREES"
50 GOTO 10
60 END
```

Problem 4

```
10 PRINT"GIVEN SECTOR ANGLE AND RADIUS, FIND ARCLENGTH"
20 PRINT"(USE PI = 3.14159)"
30 PRINT"IF SECTOR ANGLE IS IN DEGREES PRESS 1,"
40 PRINT"IN RADIAN PRESS 2";
50 INPUT X
60 IF X=1 THEN 90
70 INPUT"ENTER SECTOR ANGLE IN RADIAN";S
80 GOTO 110
90 INPUT"ENTER SECTOR ANGLE IN DEGREE";D
100 LET S=(D/180)*3.14159
110 INPUT"ENTER RADIUS OF THE CIRCLE";R
120 LET L=R*S
130 PRINT"THE ARCLENGTH = "L
140 GOTO 10
150 END
```

Problem 5

```
10 PRINT"GIVEN SECTOR ANGLE, FIND AREA OF SECTOR"
20 PRINT"(USE PI = 3.14159)"
30 PRINT"IF SECTOR ANGLE IS IN DEGREE PRESS 1,"
40 PRINT"IN RADIAN PRESS 2";
50 INPUT X
60 IF X=1 THEN 90
70 INPUT"ENTER SECTOR ANGLE IN RADIAN";S
80 GOTO 110
90 INPUT"ENTER SECTOR ANGLE IN DEGREE";D
100 LET S=(D/180)*3.14159
110 INPUT"ENTER RADIUS OF THE CIRCLE";R
120 LET A=.5*(R^2)*S
130 PRINT"AREA OF THE SECTOR = "A" SQUARE UNITS"
140 GOTO 10
150 END
```

Topic 9

Trigonometry

Problem 1

```
10 REM DETERMINE WHETHER A RIGHT ANGLE TRIANGLE
20 INPUT"ENTER VALUES FOR SIDE A, B, C";A,B,C
30 IF A^2+B^2=C^2 THEN 80
40 IF A^2+C^2=B^2 THEN 80
50 IF B^2+C^2=A^2 THEN 80
60 PRINTA" "B" "C" ARE NOT SIDES OF A RIGHT ANGLE TRIANGLE
70 GOTO 20
80 PRINT A" "B" "C" ARE SIDES OF A RIGHT ANGLE TRIANGLE"
90 GOTO 20
100 END
```

Problem 2

```
10 REM FIND SIN A, COS A, TAN A, A=30 AND 60 DEGREES
20 LET A=1: B=3 .5: C=2
30 PRINT"SIN 30 = "A/C
40 PRINT"COS 30 = "B/C
50 PRINT"TAN 30 = "A/B
60 PRINT"SIN 60 = "B/C
70 PRINT"COS 60 = "A/C
80 PRINT"TAN 60 = "B/A
90 END
```

Problem 3

```
10 REM FIND SIN A
20 LET X=3: Y=8
30 LET Z=(X^2+Y^2)^.5
40 PRINT"SIN A = "Y/Z
50 END
```

Problem 4

```
10 REM  COS A = 0.7, FIND SIN A AND TAN A
20 LET X=.7
30 LET Y=(1-X^2)^.5
40 PRINT"SIN A = "Y
50 PRINT"TAN A = "Y/X
60 END
```

Problem 5

```
10 REM CONVERT DEGREES TO RADIANS
20 PRINT"GIVE ME AN ANGLE IN DEGREE";
30 INPUT D
40 LET R=(D/180)*3.14159
50 PRINT D" DEGREES = "R" RADIANS"
60 GOTO 20
70 END
```

Problem 6

```
10 REM INPUT ANGLE (DEGREES), DISPLAY SIN A, COS A, TAN A
20 INPUT"WHAT IS YOUR DEGREE ANGLE";D
30 LET R=(D/180)*3.14159
40 PRINT"SIN "A" = "SIN(R)
50 PRINT"COS "A" = "COS(R)
60 PRINT"TAN "A" = "TAN(R)
70 GOTO 20
80 END
```

Topic 10

Statistics

Problem 1

```
10 REM THIS PROGRAM WILL PLACE NUMBERS IN CLASSES OF WIDTH
20 REM 1000 AND FIND FREQUENCY IN EACH CATEGORY AND YOU
30 REM CAN MODIFY THE PROGRAM TO FIND MEAN OF GROUPED DATA
40 REM YOU DON'T HAVE TO MAKE AN ARRAY OF THE DATA
50 INPUT"HOW MANY NUMBERS DO YOU HAVE";N
60 DIM X(N)
70 FOR I=1 TO N
80 PRINT"ENTER YOUR #";I;" NUMBER";
90 INPUT X(I)
100 NEXT I
110 INPUT"WHAT IS THE LOWEST NUMBER";A
120 INPUT"WHAT IS THE HIGHEST NUMBER";B
130 LET K=INT((B-A)/1000)+1
140 PRINT"NUMBER OF CLASSES = ";K;" (CLASS WIDTH = 1000)"
150 REM DELAY THE PROGRAM
160 FOR D=1 TO 1500:NEXT D
170 INPUT"ENTER A LOWER BOUNDARY OF THE LOWEST CLASS";M
180 LET C(J)=0
190 REM COUNT FREQUENCY FOR EACH CLASS
200 FOR I=1 TO N
210 FOR J=1 TO K
220 REM L=LOWER BOUNDARY, U=UPPER BOUNDARY
230 LET L=M+1000*(J-1): U=M+1000*(J)
240 IF L<=X(I) AND X(I)<U THEN C(J)=C(J)+1
250 NEXT J
260 NEXT I
270 PRINT"=========================="
280 PRINT"    CLASS      FREQUENCY"
290 PRINT"=========================="
300 FOR J=1 TO K
310 PRINT L;"-";U;"    ";C(J)
320 NEXT J
330 PRINT"=========================="
340 PRINT"           TOTAL  ";N
350 PRINT"=========================="
360 END
```

Problem 2

```
10 REM FIND MEAN OF ANY GIVEN SERIES OF NUMBERS
20 PRINT"HOW MANY NUMBERS DO YOU HAVE";N
30 INPUT N
40 LET Y=0
50 FOR I=1 TO N
60 PRINT"ENTER YOUR #"I" NUMBER";
70 INPUT X(I)
80 LET Y=Y+X(I)
90 NEXT I
100 PRINT"MEAN = "Y/N
110 GOTO 20
120 END
```

Problem 3

```
10 REM FIND MEDIAN OF ANY GIVEN 5 NUMBERS
20 PRINT"ENTER 5 NUMBERS IN ASCENDING OR DESCENDING ORDER"
30 FOR I=1 TO 5
40 PRINT"ENTER YOUR #"I" NUMBER";
50 INPUT X(I)
60 NEXT I
70 PRINT"THE MEDIAN IS "X(3)
80 GOTO 20
90 END
```

General Program for Problem 3

```
10 REM THIS PROGRAM FIND MEDIAN OF ANY GIVEN NUMBERS
20 REM WITHOUT SORTING THE NUMBERS
30 INPUT"HOW MANY NUMBERS DO YOU HAVE";N
40 FOR I=1 TO N
50 PRINT"ENTER YOUR #"I" NUMBER";
60 INPUT X(I)
70 NEXT I
80 REM THIS LOOP SORTING THE NUMBERS
90 FOR I=1 TO N-1
100 FOR J=I+1 TO N
110 IF X(I)<X(J) THEN 150
120 LET X=X(I)
130 LET X(I)=X(J)
140 LET X(J)=X
150 NEXT J
160 NEXT I
170 IF N/2=INT(N/2) THEN 200
180 LET M=INT(N/2)+1
190 PRINT"THE MEDIAN = "X(M): GOTO 30
200 LET M=X(N/2)+X(1+N/2)
210 PRINT"THE MEDIAN = "M/2: GOTO 30
220 END
```

Problem 4

```
10 REM DETERMINE PERCENTILE
20 INPUT"HOW MANY STUDENTS IN THE CLASS";N
30 INPUT"WHAT IS YOUR RANK";R
40 LET A=((R-1)*100)/N
50 PRINT"THE PERCENTILE IS "A
60 END
```

Topic 11

Quadratic Functions

Problem 1 & 2

```
10 REM FIND AXIS OF SYMMETRY, VERTEX, POINT OF
20 REM INTERSECTION WITH Y-AXIS OF Y=A(X^2)+BX+C
30 INPUT"ENTER VALUES FOR A, B, C";A,B,C
40 LET X=-B/(2*A)
50 LET Y=-(B*B-4*A*C)/(4*A)
60 PRINT"THE AXIS OF SYMMETRY IS X = "X
70 PRINT"THE POINT OF VERTEX IS ("X","Y")"
80 PRINT"GRAPH INTERSECTS Y-AXIS AT (0,"C")"
90 GOTO 30
100 END
```

Problem 3

```
10 REM FIND MAX. AND MIN. VALUE OF Y=A(X^2)+BX+C
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 LET X=-B/(2*A)
40 LET Y=-(B*B-4*A*C)/(4*A)
50 IF A<0 THEN 80
60 PRINT"FUNCTION HAS A MIN. VALUE OF "Y" WHEN X = "X
70 GOTO 90
80 PRINT"FUNCTION HAS A MAX. VALUE OF "Y" WHEN X = "X
90 END
```

Problem 4

```
10 PRINT"DETERMINE ROOTS OF A(X^2)+BX+C = 0"
20 INPUT"ENTER VALUES FOR A, B, C";A,B,C
30 LET D=(B*B)-(4*A*C)
40 IF D<0 THEN PRINT"EQUATION HAS NO REAL ROOTS":GOTO 10
50 LET X=-B/(2*A)
60 IF D=0 THEN PRINT"EQUATION HAS ONE REAL ROOT:X = "X:
                                                 GOTO 10
70 LET X1=(-B+(D^.5))/(2*A):X2=(-B-(D^.5))/(2*A)
80 PRINT"EQUATION HAS TWO REAL ROOTS: X1="X1", X2="X2
90 GOTO 10
100 END
```